

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



### THESIS

#### FRAMEWORK FOR A LINK LAYER PACKET FILTERING (LLPF) SECURITY PROTOCOL

by  
Gregorio G. Darroca

September 1998

Thesis Advisors:

Associate Advisor:

Geoffrey Xie  
Cynthia Irvine  
Rex Buddenberg

Approved for public release; distribution is unlimited.

19981103 054

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 1998		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE FRAMEWORK FOR A LINK LAYER PACKET FILTERING (LLPF) SECURITY PROTOCOL			5. FUNDING NUMBERS	
6. AUTHOR(S) Gregorio G. Darroca				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) Transport Layer (OSI Layer 3) switching and routing provides routing flexibility but not high throughput. Link layer (OSI Layer 2) switching provides high throughput but not the routing flexibility needed to manage topology change and load fluctuations in the network. Neither Layer 3 routing nor Layer 2 switching protocols were originally designed to support confidentiality and integrity of data, and authentication of participants. Proposals to integrate security may have positive results for data confidentiality, integrity and authentication, but often result in additional overhead, increased transmission latency, and decreased throughput. An added difficulty is reconciling standards and protocols when integrating heterogeneous routing networks with homogenous switching networks while minimizing impact on throughput. This thesis examined current Internet extensions and architectures as well as IP security services and Layer 2 switching in IP-based networks. Requirements for a framework for a proposed security protocol include: Link Layer switching and routing; independence of particular communication protocols and standards; IP packet filtering and routing according to predetermined security policies and with no significant impact on throughput; and continued routing flexibility of IP. This security protocol, called Link Layer (Link Layer Packet Filtering (LLPF)), filters packets at the Link Layer, and boasts two innovations: use of an authentication trailer and multiple cryptographic keys with short cryptoperiods.				
14. SUBJECT TERMS network security, Asynchronous Transmission Mode (ATM), internetworking, protocol			15. NUMBER OF PAGES 208	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFI- CATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18  
298-102



Approved for public release; distribution is unlimited

**FRAMEWORK FOR A LINK LAYER PACKET FILTERING (LLPF)  
SECURITY PROTOCOL**

Gregorio G. Darroca  
B.S., U.S. Naval Academy, 1979

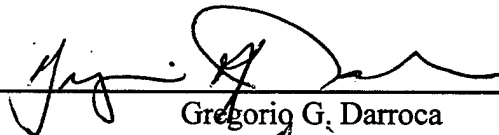
Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY  
MANAGEMENT**

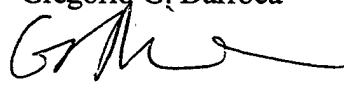
from the

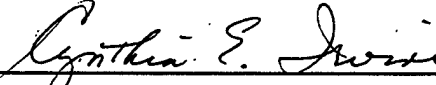
**NAVAL POSTGRADUATE SCHOOL  
September 1998**


Author:

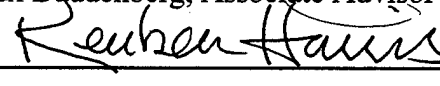
  
Gregorio G. Darroca

Approved by:

  
Geoffrey Xie, Thesis Advisor

  
Cynthia Irvine, Thesis Advisor

  
Rex Buddenberg, Associate Advisor

  
Reuben Harris, Chairman  
Department of Systems Management



## ABSTRACT

Transport Layer (OSI Layer 3) switching and routing provides routing flexibility but not high throughput. Link layer (OSI Layer 2) switching provides high throughput but not the routing flexibility needed to manage topology change and load fluctuations in the network. Neither Layer 3 routing nor Layer 2 switching protocols were originally designed to support confidentiality and integrity of data, and authentication of participants. Proposals to integrate security may have positive results for data confidentiality, integrity and authentication, but often result in additional overhead, increased transmission latency, and decreased throughput. An added difficulty is reconciling standards and protocols when integrating heterogeneous routing networks with homogenous switching networks while minimizing impact on throughput.

This thesis examined current Internet extensions and architectures as well as IP security services and Layer 2 switching in IP-based networks. Requirements for a framework for a proposed security protocol include: Link Layer switching and routing; independence of particular communication protocols and standards; IP packet filtering and routing according to predetermined security policies and with no significant impact on throughput; and continued routing flexibility of IP. This security protocol, called Link Layer (Link Layer Packet Filtering (LLPF)), filters packets at the Link Layer, and boasts two innovations: use of an authentication trailer and multiple cryptographic keys with short cryptoperiods.



# TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
A.	EXECUTIVE SUMMARY .....	1
B.	PROBLEM DEFINITION .....	4
C.	MOTIVATION .....	6
D.	THESIS OBJECTIVE.....	7
E.	THESIS STRUCTURE.....	8
	1. Scope.....	8
	2. Organization.....	8
<b>II.</b>	<b>BACKGROUND AND RELATED WORK .....</b>	<b>11</b>
A.	INTRODUCTION .....	11
B.	INFORMATION SECURITY .....	11
C.	OSI LAYER 2 (LINK LAYER) SWITCHING.....	15
	1. Asynchronous Transmission Mode (ATM).....	15
	a. <i>ATM Cell Format</i> .....	16
	b. <i>ATM Switch Operations</i> .....	19



<i>c. Layers</i> .....	25
<i>d. Signaling</i> .....	25
<i>e. Technology</i> .....	27
2. Classical Internet Protocol (IP) Over ATM.....	28
D. OSI LAYER 3 (NETWORK LAYER) PROPOSALS .....	31
1. Security Architecture for IP (IPsec) Protocol .....	31
<i>a. Security Policy Database (SPD)</i> .....	32
<i>b. Operation</i> .....	33
<i>c. Key Management</i> .....	33
<i>d. Security Association Database (SAD)</i> .....	39
<i>e. Performance Issues</i> .....	40
2. Authentication Header Protocol.....	41
<i>a. Format</i> .....	41
<i>b. Operations</i> .....	42
3. Encapsulating Security Payload (ESP) Protocol.....	48
<i>a. Syntax</i> .....	49

<i>b. Algorithms</i> .....	52
<i>c. Transport Mode</i> .....	52
<i>d. Tunnel Mode</i> .....	53
<i>e. Outbound Processing</i> .....	54
<i>f. Inbound Processing</i> .....	55
4. Flow Based Security Protocol.....	56
<i>a. Description</i> .....	56
<i>b. Datagram Semantics</i> .....	56
<i>c. Flow</i> .....	57
<i>d. Protocol Overview</i> .....	57
<i>e. Protocol Operation</i> .....	61
E. BENEFITS .....	62
F. TAG SWITCHING.....	63
1. Tag Edge Routers.....	63
2. Tag Switches.....	65

3. Tag Distribution Protocol (TDP) .....	66
G. SUMMARY .....	66
<b>III. FRAMEWORK FOR A LINK LAYER PACKET FILTERING (LLPF) SECURITY PROTOCOL .....</b>	<b>71</b>
A. INTRODUCTION .....	71
B. THE CONCEPT .....	74
1. Header VS Trailer .....	74
2. In-Band VS Out-of-Band Call Setup and Connection Management ...	75
3. Cryptographic Key Management .....	77
4. Authentication Trailer and IP Routing .....	78
5. Filtering Technique .....	80
6. Authentication and Filtering Servers .....	84
a. <i>Master Authentication Server (MAS)</i> .....	84
b. <i>Packet Filtering Gateway Server (PFGS)</i> .....	85
7. Cryptographic Key table .....	85
8. Packet Fragmentation .....	88

C.	OPERATION .....	89
D.	KEY UTILIZATION AND MANAGEMENT .....	91
E.	POLICY .....	92
F.	INTEROPERABILITY.....	92
G.	SUMMARY .....	93
<b>IV.</b>	<b>CONCLUSIONS AND RECOMMENDATIONS.....</b>	<b>95</b>
A.	INTRODUCTION .....	95
B.	RESEARCH CONCLUSIONS.....	95
1.	IP-Based Security Approach.....	95
2.	ATM and Tag Switching .....	96
3.	Link Layer Packet Filtering (LLPF) Security Protocol .....	97
C.	RECOMMENDATIONS FOR FUTURE WORK .....	98
D.	SUMMARY .....	100
APPENDIX A.	ISO OSI REFERENCE MODEL .....	101
APPENDIX B.	INTERNET PROTOCOL VERSION 4.....	113
APPENDIX C.	INTERNET PROTOCOL VERSION 6.....	133

APPENDIX D.	TRANSPORT CONTROL PROTOCOL (TCP).....	159
LIST OF REFERENCES .....		179
BIBLIOGRAPHY .....		181
INITIAL DISTRIBUTION LIST .....		185

## LIST OF FIGURES

<b>Figure 1.</b>	ATM cell format .....	17
<b>Figure 2.</b>	ATM cell header format.....	17
<b>Figure 3.</b>	System view of ATM process After (Wu, 1998).....	20
<b>Figure 4.</b>	Virtual path and virtual channel From (Cisco, 1996) .....	23
<b>Figure 5.</b>	Switching table and cell routing .....	24
<b>Figure 6.</b>	ATM funtional elements mapped to OSI layers From (Cisco, 1998).....	24
<b>Figure 7.</b>	System view of classical IP model.....	29
<b>Figure 8.</b>	Cell preparation From (Cabletron, 1997).....	31
<b>Figure 9.</b>	Combining of security associations From (Kent and Atkinson, 1998).....	35
<b>Figure 10.</b>	End-to-end security between 2 hosts From (Kent and Atkinson, 1998)...	36
<b>Figure 11.</b>	Simple VPN From (Kent and Atkinson, 1998).....	36
<b>Figure 12.</b>	Combination of end-to-end and VPN From (Kent and Atkinson, 1998)..	37
<b>Figure 13.</b>	Remote/Mobile host and VPN From (Kent and Atkinson, 1998) .....	37
<b>Figure 14.</b>	Same SA endpoints From (Kent and Atkinson, 1998) .....	38
<b>Figure 15.</b>	One endpoint is the same From (Kent and Atkinson, 1998).....	38

<b>Figure 16.</b>	End points are different From (Kent and Atkinson, 1998) .....	39
<b>Figure 17.</b>	AH format From (Kent and Atkinson, 1998).....	41
<b>Figure 18.</b>	IPv4 before & after AH applied From (Kent and Atkinson, 1998) .....	46
<b>Figure 19.</b>	IPv6 before & after AH applied From (Kent and Atkinson, 1998) .....	46
<b>Figure 20.</b>	IPv4 & IPv6 authenticated fields .....	47
<b>Figure 21.</b>	ESP header elements From (Kent and Atkinson, 1998) .....	50
<b>Figure 22.</b>	ESP header transport mode application to IP packet From (Kent and Atkinson, 1998).....	53
<b>Figure 23.</b>	ESP header tunnel mode application to IPpacket From (Kent and Atkinson, 1998).....	54
<b>Figure 24.</b>	Flow state table After (Mittra and Woo, 1997).....	59
<b>Figure 25.</b>	Zero-based keying mechanism .....	59
<b>Figure 26.</b>	FBS header From (Mittra and Woo, 1997).....	60
<b>Figure 27.</b>	FBS protocol architecture and Operation From (Mittra and Woo, 1997).	62
<b>Figure 28.</b>	Tag switching network.....	64
<b>Figure 29.</b>	System model for design of LLPF .....	73
<b>Figure 30.</b>	Encapsulation of IP packet with authentication trailer (AT) .....	80
<b>Figure 31.</b>	Aunthentication Trailer format .....	82

<b>Figure 32.</b>	Tunneled IP packet with AT .....	82
<b>Figure 33.</b>	A key with index .....	87
<b>Figure 34.</b>	Sample key table .....	87
<b>Figure 35.</b>	Communication between two computer systems.....	103
<b>Figure 36.</b>	Relationship between adjacent layers in a single system.....	104
<b>Figure 37.</b>	Headers and data .....	106
<b>Figure 38.</b>	Protocol relationships.....	116
<b>Figure 39.</b>	Transmission path .....	118
<b>Figure 40.</b>	Gateway protocols .....	122
<b>Figure 41.</b>	Example of Internet datagram header .....	123
<b>Figure 42.</b>	Type-of-Service .....	125
<b>Figure 43.</b>	Various control flags.....	127



## LIST OF TABLES

<b>Table 1.</b> Traffic classes supported by ATM adaptation layer .....	23
<b>Table 2.</b> Functions of ATM layers .....	26
<b>Table 3.</b> IPv4 & IPv6 Mutable/Immutable fields.....	45

## LIST OF ACRONYMS

AAL	ATM Adaption Layer
AH	Authentication Header
AT	Authentication Trailer
ATM	Asynchronous Transmission Mode
BGP	Border Gateway Protocol
DOS	Denial-of-Service
CLP	Cell Loss Priority
CoS	Class of Service
CPI	Common Part Indicator
CRC	Cyclic Redundancy Check
CS	Convergence Sublayer
DARPA	Defense Advance Research Projects Agency
DES	Digital Encryption System
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ESP	Encapsulating Security Payload
FAM	Flow Association Mechanism
FBS	Flow Based Security
FTP	File Transfer Protocol
GFC	Generic Flow Control

HEC	Header Error Control
HMAC	Hashing for Message Authentication Code
ICMP	Internet Control Message Protocol
ICV	Integrity Check Value
ISO	International Standards Organization
IEEE	Institute of Electrical & Electronics Engineers
IGMP	Internet Group Management Protocol
IKE	Internet Key Exchange
IPL	Inner Packet Length
IP	Internet Protocol
IPsec	Internet Protocol security architecture
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ITU-T	International Telecommunications Union- Telecommunication Standard Section
LAN	Local Area Network
LLC	Logical Link Control
LLPF	Link Layer Packet Filter
MAC	Message Authentication Code
MAS	Master Authentication Server
MCR	Minimum Cell Rate
MD5	Message Digest 5

MTU	Maximum Transmission Unit
OSI	Open Systems Interconnection
PCR	Peak Cell Rate
PFGS	Packet Filter Gateway Server
PKI	Public Key Infrastructure
PDU	Protocol Data Unit
PVC	Permanent Virtual Channel
QoS	Quality of Service
RFC	Request-for-Comments
RFKC	Receive Flow Key Cache
RPC	Remote Procedures Call
SA	Security Association
SAD	Security Association Database
SAR	Segmentation and Reassembly
SDU	Service Data Unit
SCR	Sustained Cell Rate
SFL	Security Flow Label
SHA	Secure Hashing Algorithm
SHS	Secure Hash Standard
SKIP	Simple Key Management for Internet Protocol
SMDS	Switched Multimegabit Data Service
SNAP	Subnetwork Attachment Point

SNMP	Simple Network Management Protocol
SPI	Security Parameter Index
SPD	Security Policy Database
SONET	Synchronous Optical Network
TCP	Transport Control Protocol
TDP	Tag Distribution Protocol
TER	Tag Edge Router
TDM	Time Division Multiplexing
TFKC	Transmission Flow Key Cache
TIB	Tag Information Base
UDP	User Datagram Protocol
VC	Virtual Connection
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier
VPN	Virtual Private Network
ZKM	Zero Keying Mechanism

# **I. INTRODUCTION**

## **A. EXECUTIVE SUMMARY**

Confronted with the growing demand for integrated transport of multirate and multimedia traffic, the international telecommunication community developed a series of specifications which now form the core of the OSI link layer (Layer 2)-based switching technology called Asynchronous Transmission Mode (ATM). The ATM Forum, an international non-profit organization, guided the convergence of these interoperability specifications in order to accelerate the use of ATM products and services. The ATM technology is specifically designed to support high-speed digital voice and data communications. Today, ATM switches form the backbone for telecommunications in the United States. Even the Internet which relies on its multitude of routers (OSI Network layer routing, Layer 3) still utilizes an inner core of ATM switches to function. Layer 3 routing provides the routing flexibility but not the throughput. In contrast, Layer 2 switching provides the throughput but not the routing flexibility to packet or cells to manage topology change and load fluctuations in the network. One other crucial aspect of telecommunication, which neither paradigm addresses, is security of data. As originally proposed, neither Layer 3 routing nor Layer 2 switching protocols carried features or design accommodations for confidentiality and integrity of data during transmission, and authentication of data origin. Attempts and proposals to integrate security may have positive results for data confidentiality, integrity and authenticity, but often result in additional overhead, causing data throughput to decrease. Confounding the situation is the difficulty of reconciling standards and protocols when integrating

largely heterogeneous routing networks (primarily connectionless mode Layer 3) with homogenous switching networks (primarily connection mode Layer 2) while minimizing the impact on throughput.

As can be surmised, security in a telecommunication network exacerbates the existing tension between throughput and routing flexibility. Currently proposed network security architectures for Layer 3 such as IPsec, Authentication Header (AH) and Encapsulating Security Payload (ESP) do provide confidentiality, authentication and integrity services, and routing flexibility. However, their penalty on throughput and latency is so significant that their successful application is limited to text data only. So the challenge is to develop a security strategy that would provide the maximum security of transmitted information in a complex network, such as the Internet, while maintaining high data throughput and routing flexibility.

Our proposed solution is a framework for a network protocol that provides security services while maintaining high throughput and routing flexibility. The framework is a packet filtering approach and has the following characteristics:

- o Security services (provided at the packet level) include data origin authentication and data integrity
- o Uses an Authentication Trailer (AT) appended to each packet, thus avoiding the overhead (latency) associated with the processing of complex IP headers
- o Processing of security data is conducted at the Link Layer, thus maintaining high throughput

- o It uses short duration cryptographic keys (with less complex one-way hash functions), thus greatly minimizing successful brute force attacks on captured packets
- o It utilizes an automated key management scheme that supplies participating hosts with multiple session keys
- o It provides flexibility in the selection of cryptographic algorithms by the user
- o It uses the IP tunneling technique
- o It remains compatible with current Internet standards and protocols and switching technology

The framework for the Link Layer Packet Filtering (LLPF) security protocol integrates the features that best meet security with high throughput and routing flexibility, from Tag Switching, FBS, IPsec, and the Internet Protocol. However, LLPF boasts two innovations that set it apart from the security solutions described in Chapter II. These innovations are the use of an authentication trailer and multiple session keys of short duration.

In addition to data origin authentication and data integrity, LLPF security protocol is more capable of handling Denial-of-Service (DOS) attacks due to its high throughput capability. DOS attacks such as SYN storm, UDP bomb, Finger bomb, Data flood, Echo-and-Chargen Check, Log flood and Open Close do not have the ability to penetrate nor



the capacity to overwhelm an LLPF compliant gateway server that can filter packets at gigabit speed.

## **B. PROBLEM DEFINITION**

The public switched network is the heart of telecommunications in the United States and the world. During the 1960's and 70's, the ever increasing demands in telecommunications for bandwidth and higher throughput motivated the upgrade of public switched telephone systems in the U.S. from all-analog systems to networks supporting a combination of analog and digital requirements. Today, digital communication has proven to be more reliable, more scalable, and of increased quality compared to its analog counterpart. Digital communication includes support for data and video as well as voice.

Confronted with the growing demand for integrated transport of multirate and multimedia traffic, the international telecommunication community developed a series of specifications which now form the core of the OSI link layer (Layer 2)-based switching technology called Asynchronous Transmission Mode (ATM). The ATM Forum, an international non-profit organization, guided the convergence of these interoperability specifications in order to accelerate the use of ATM products and services. The ATM technology is specifically designed to support high-speed digital voice and data communications. Today, ATM switches form the backbone for telecommunications in the United States. Even the Internet which relies on a multitude of routers (OSI Network layer routing, Layer 3) still utilizes an inner core of ATM switches to function. Layer 3 routing provides the routing flexibility but not the throughput. In contrast, Layer 2

switching provides the throughput but not the routing flexibility to packet or cells during network peak and down periods. One other crucial aspect of telecommunication, which neither paradigm addresses, is security of data. As initially proposed, neither Layer 3 routing nor Layer 2 switching protocols carried features or design accommodations for confidentiality and integrity of data during transmission, and authentication of data origin. Attempts and proposals to integrate security may have positive results for data confidentiality, integrity and authenticity, but are likely to result in additional overhead, causing data throughput to decrease. Confounding the situation is the difficulty of reconciling standards and protocols when integrating largely heterogeneous routing networks (primarily connectionless mode Layer 3) with homogenous switching networks (primarily connection mode Layer 2) while minimizing the impact on throughput.

As can be surmised, security in a telecommunication network has a negative influence with respect to throughput and routing flexibility, in addition to the existing tension between throughput and routing flexibility. Currently proposed network security architectures for Layer 3 such as IPsec, Authentication Header (AH) and Encapsulating Security Payload (ESP) do provide confidentiality, authentication and integrity services, and routing flexibility. However, their penalty on throughput is so significant that successful application is limited to text data only. So the challenge is to develop a security strategy that would provide the maximum possible protective services to transmitted information in a complex network, such as the Internet, while maintaining high data throughput and routing flexibility.

## C. MOTIVATION

Tag Switching, proposed by CISCO Systems (Rechter, Davie et al., 1997) is designed to provide flexibility and added functionality in Layer 3 routing, and can be readily adapted for Layer 2 switching. Consisting primarily of two components, forwarding and control, Tag Switching uses the notion of label swapping. Specifically, packets having the same final destination, destined for the same output port in a switch, or sharing a virtual path to the destination, are tagged in the header with a fixed length, fairly short labels. These labels (also referred as tags) are indices in an exact-match algorithm intended to simplify packet forwarding procedure, which in turn enables higher forwarding performance and allows straightforward hardware implementation. However, Tag Switching is optimized for throughput performance and such performance is achieved by compromising in other functional areas. In particular, the “cut-through” routing used by Tag Switching leaves it vulnerable to various network attack techniques.

The Flow-Based Security (FBS) protocol, proposed by Suvo Mittra of Stanford University and Thomas Woo of Bell Laboratories, relies on datagram semantics and the concept of flows for achieving routing flexibility. Security is based on zero-message keying and soft state processing on a per-packet basis. The flow paradigm ensures data integrity in multimedia and multi-user sessions, while security processing on a per-packet basis provides protection against total compromise of a flow should one packet key be compromised. Additionally, FBS provides an adequate countermeasure to replay attacks by using timestamps on packets. The timestamps are adjusted with “freshness windows” to account for transmission delays and unsynchronized machines. FBS can be

implemented using the Public Key Infrastructure (PKI) system for authentication of users and encryption of the data field. The weakness of FBS is that the implementation is still restricted to OSI Layer 3 and above, thereby inheriting the throughput deficiency of the OSI Network Layer.

Although none of the above offers a combined solution of network security, high throughput and routing flexibility, each proposal does offer a feasible solution to one or two of the three significant requirements for seamless, secure area network connectivity. Tag Switching is a superior approach to diminishing the penalties in throughput when transitioning from a largely heterogeneous IP/Layer 3 routing network to a homogenous ATM/Layer 2 switching network. It utilizes existing Internet routing protocol/standards, thus routing flexibility is preserved at the heterogeneous portion of the path of travel, and modifications involved are limited to integrating the Tag Switching protocol into participating routers and switches. However, Tag Switching completely neglects security, which is a primary focus of FBS. In addition to security, FBS maintains the routing flexibility afforded by datagram semantics, but it conducts the mechanics of security and routing at Layer 3. As a result, FBS is unable to approach the throughput benefits of Tag Switching.

#### **D. THESIS OBJECTIVE**

The objective of this thesis is to develop a framework for a security protocol that will seamlessly integrate with fast IP OSI layer 2 switching. The research will include examining/evaluating network security architectures such as FBS and other IP based architecture (IPsec protocol, AH & ESP protocols), and explore the integration of

proposed protocols with Layer 2 IP packet forwarding technique such as Tag Switching.

This work will form the basis for a security protocol having the following characteristics:

- o Operate/function no higher than the data link layer,
- o Not cause significant decrease of packet throughput,
- o Not be dependent on one particular communication protocol/standard,
- o Filter/route IP packets according to predetermined security policies, and
- o Maintain routing flexibility of Layer 3.

## **E. THESIS STRUCTURE**

### **1. Scope**

This thesis will focus on network security architectures that would include the notion of flows in a datagram network, and explore the integration of flows with Layer 2 packet/cell forwarding techniques such as ATM. The resulting design will then be examined to determine the feasibility of providing effective security while maintaining high data throughput and routing flexibility.

### **2. Organization**

The introductory chapter characterizes the general problem and explains the motivation for this research project. Chapter II discusses relevant information describing existing network security protocols for the Internet, a recently proposed flow-based security architecture and the proposed Internet protocol to interface the IP format with

OSI Layer 2 switching such as ATM. Chapter III presents a framework for a security protocol compatible with OSI Layer 2 switching. The thesis project conclusions and recommendations for further work are presented in Chapter IV.



## **II. BACKGROUND AND RELATED WORK**

### **A. INTRODUCTION**

This chapter discusses background information on network security and proposals on establishing secure communication sessions through public networks such as the Internet. Section B summarizes the security services that must be provided in order to protect it from a variety of threats. Section C describes the ATM protocol and the interface between IP and ATM. A description of proposed security architecture for OSI Layer 3 based protocol (IP) and its two primary elements is provided in section D. In addition, datagram security architecture based on the notion of flow is described. Section E concludes the chapter with a discussion of the strengths and weaknesses of the current transport technology and security protocols and why the tension between security and throughput remains unresolved.

### **B. INFORMATION SECURITY**

In the past, information security of an organization was focused on two major areas: Physical security and personal reliability (Stallings 1998). The latter was achieved by using material and procedures (safes, cipher lock doors, guarded buildings, fenced grounds, etc.) to control and restrict access to sensitive information. The former was accomplished by personnel screening/background investigation procedures. Today this remains true with current security practices in the government and the military services. According to Stallings (Stallings 1998), information security requirements of any organization of today have experienced two major changes that have influenced its



paradigm of security practices. First is the widespread utilization of computers in conducting daily businesses, and second, the networking of these computers to communicate and share information in a distributed manner. Information that was on paper documents now exists in electronic form, which can be copied multiple times with or without the knowledge and authorization of the originator. The ability to communicate and share information via the computer/network model inspite of distances between users has introduced significant uncertainty in the true identities of users and the authenticity of information. Security requirements remain as they were, but the methodologies of execution now go beyond the immediate physical and personal focus. The computer/network model has added another dimension to information security.

Physical security was practiced primarily on the premise that the information requiring protection exists on a document (paper). The computer/network model stores, manipulates and displays information in electronic format. Accordingly, security services must now reflect provisions in affording information security to such a medium. The following list is generally accepted as classifications of information security services to the computer/network model:

- o Confidentiality - protecting the information in a computer or transiting through a network from access/reading by unauthorized parties.
- o Authentication - obtaining assurance that the party wishing to communicate and/or the party responding to the communication request is truthfully stating their identity. In the case of data, authentication assures the recipient that it is

from the origin it claims to be. Examples of authentication protocols and services include, for example: MAC/HMAC, SHA, MD5, PKI, Kerberos (Stallings 1998).

- o Integrity - ensures that any modification (writing, changing status, deleting, creating, and replaying) of computer system assets and transmitted data are conducted by authorized parties only, (methods for achieving network communications integrity include the use of timestamps and techniques cited for authentication above).

- o Nonrepudiation - not allowing the sender or receiver of information to deny the transmission. Nonrepudiation services are provided by such methods as DSS and DSA (Stallings 1998)

- o Access control - controlling access to the information resources by or for authorized parties. Access control is used to ensure confidentiality, integrity, and, to some extent, availability of resources.

- o Availability - requiring authorized parties to have access on demand to information resources.

Threats posed to digital information and resources come in two categories:

Passive and active. Eavesdropping or monitoring network traffic characterizes passive attacks. If the information intercepted during an attack is unprotected (i.e., unencrypted), then the attacker has full access to the details/contents. If the information intercepted is protected (i.e., encrypted), then it is subject to traffic analysis attack. Successful traffic

analysis of intercepted information reveals intent or future actions without knowing the actual content of the intercepted traffic. Passive attacks, by their very nature, are much more difficult to detect.

Active attacks are overt and intrusive in nature. They are characterized by modifying contents in data stream or insertion of false data in the stream. Active attacks come in four categories:

- o Masquerade - impersonation of an authorized entity by an unauthorized party. If both parties are authorized, then masquerade describes the impersonation of one party with greater privileges by another party with lesser privileges.
- o Replay - the interception and subsequent retransmission of data to produce a desired effect for an unauthorized party.
- o Modification - the interception and alteration of legitimate data by an unauthorized party in order to induce delay, reordering, or any other desired effect for the unauthorized third party to the data stream.
- o Denial-of-service - the disruption of a network or a portion of the network, to the point where the normal use or management of the network resources is no longer available to the authorized users.

## C. OSI LAYER 2 (LINK LAYER) SWITCHING

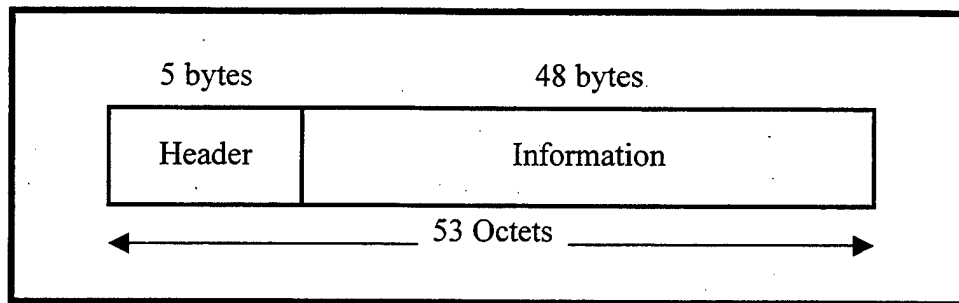
### 1. Asynchronous Transmission Mode (ATM)

Asynchronous Transmission Mode is a connection-oriented high speed, low delay Layer 2 switching technology using short, fixed-size packets called cells. It was selected in 1988 by the International Telecommunications Union (ITU) as the switching and multiplexing technique for the Broadband Integrated Services Digital Network (B-ISDN). ATM is a transport technology for all types of data (text, voice, video, image, etc.), and for a wide variety of lower layer services (Frame Relay, Switched Multimegabit Data Service (SMDS), and circuit emulation). The *asynchronous* nature of the technology refers to its non-periodic transmission (bursty traffic types) of the data that is being transmitted across an ATM network. Primarily, the asynchronous label refers to voice and video data.

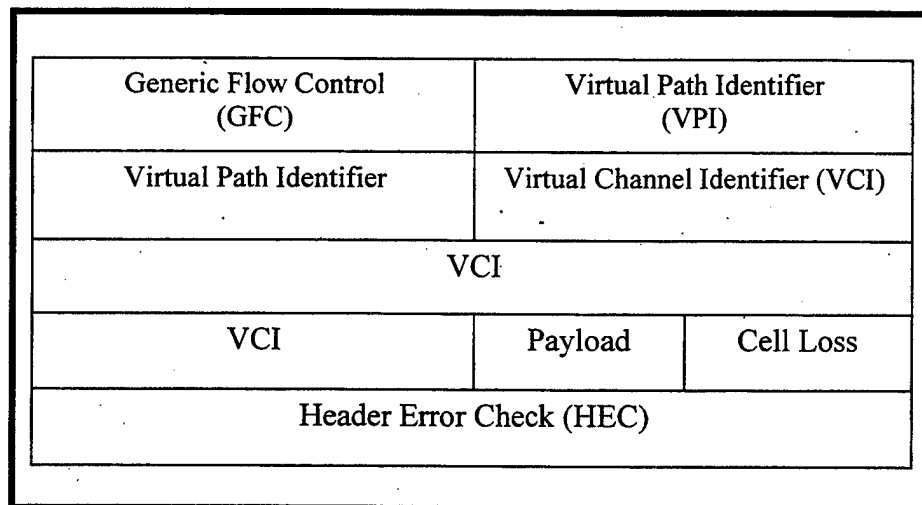
In ATM, the inefficiencies of dividing bandwidth into dedicated, fixed-size channels (e.g., TDM) for a number of connections is avoided by dynamically sharing bandwidth among multiple logical connections. Cells are transmitted one by one, and when transmitting, ATM uses the entire bandwidth. Each cell (payload capacity) contains an address to associate it with a particular logical connection. When a connection demands an increase in bandwidth, ATM simply transmit more cells for the connection. When a connection is idle, i.e., it has no cells in the network, the bandwidth is made available to other connections that needs it. ATM delivers the cells at several two standard bit rates including 155.520 Mbps and 622.080 Mbps.

***a. ATM Cell Format***

The format of an ATM cell (shown in Figure 1) consists of 48 octets (bytes) of information and 5 octets of address header, thus a total length of 53 octets per cell. The cell header construction (shown in Figure 2) is as follows:



**Figure 1.** ATM cell format



**Figure 2.** ATM cell header format

Generic Flow Control - 4-bit field end-to-end flow control. Supports both point-to-point and point-to-multipoint connection.

Virtual Path Identifier - 8-bit field routing address.

Virtual Channel Identifier - 16-bit field routing address.

Payload Type - 3-bit field describing the type of data in the information field. Can also be used to carry inband control information.

Cell Loss Priority - 1-bit field providing cell handling guidance to the network in event of congestion. A set bit ("1") indicates the cell is subject to discard, and a "0" bit indicates sufficient network resources must be allocated to preserve forwarding of the cell.

Header Error-Control (HEC) - 8-bit field used to detect single and double bit errors, and correct single bit errors in the header. Layer One processes the HEC.

The small, fixed-size cells offer several advantages over large, variable-size packets\*:

- o Less transmission time, thus reducing queueing delay and making the performance more predictable for high priority (real-time) data at a nonpreemptive transmission link.

- o Due to above, lower end-to-end network latency for real time (e.g., voice) data.

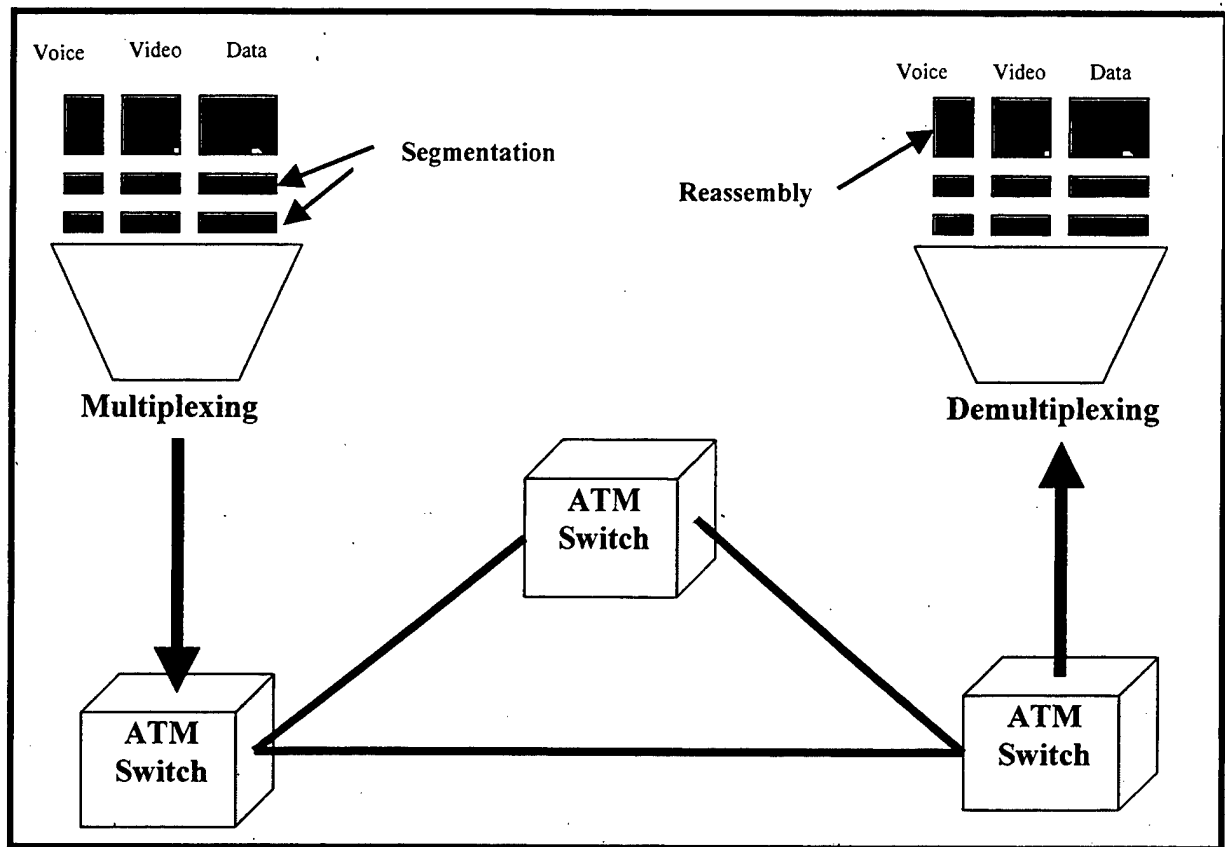
- o Fixed-sized cells can be switched more efficiently in hardware (Stallings 1992), thus very high data rate can be achieved.

\*Note: The hardware technology advances of today have made these advantages much less obvious

### ***b. ATM Switch Operations***

Upon receipt of the user's information, the ATM switch divides it into 53 bytes cells. This cell conversion is transparent to the connected equipment. The user's information can be of any type - video, data, voice, imagery, etc. If multiple connections are active within the ATM switch, it multiplexes the cells together into a single bit stream and transports the multiplexed bit stream over the physical transmission path (Layer 1: T1, T3, SONET, etc.). At each ATM switch along the transit path, the bit stream may be demultiplexed in order to comply with the routing orders of each cell. At the switch, the cells are again multiplexed into a single bit stream for the travel to the next ATM switch and/or final destination. Upon arrival at the final destination, the bit stream is demultiplexed the final time, and the appropriate adaptation processes convert the cells back into their native formats. Last, the resulting information is delivered in sequence to the upper layer protocols (e.g., FTP) for further processing (see Figure 3)





**Figure 3.** System view of ATM process  
After (Wu 1998)

Outside of header error detection and correction (executed in the physical layer), ATM relegates to the upper layer protocols (above Layer 2) the checking and correction of errors in the information payload. This approach allows ATM to avoid the significant delay/latency associated with error detection and correction in the information payload. Although in general, ATM does not care what type of traffic it is transmitting, it can discriminate and service cells accordingly based on the information contained in the ATM cell header. For obvious reasons, delay must not vary very much for cells carrying voice and video information. On the other hand, some data cells may be sensitive to loss of cells but can tolerate delays. Therefore ATM affords voice and video traffic priority with fixed delay, while concurrently ensuring that data traffic has low loss rates. To support such differences, ATM has created Traffic Classes to organize the type of service according to delivery requirements of the information (see Table 1). These categorization of services are established during the subscription period for the ATM service or during the connection set up of ATM connections. In addition to Traffic Classes, each virtual connection has parameters that specifies amount of bandwidth, priority, Quality of Service (QoS) (minimum cell rate (MCR), sustained cell rate (SCR), peak cell rate (PCR)).

There are two virtual circuit communication services in ATM: Switched Virtual Circuits (SVC) and Permanent Virtual Circuits (PVC). SVCs are short-term connections that require call setup and teardown procedures while PVCs are permanently dedicated connections, much like dedicated private lines.

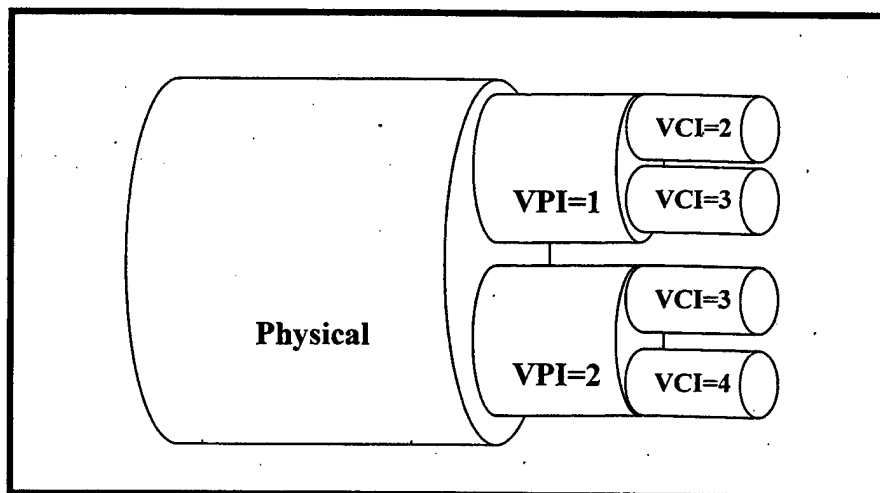
The ATM cell format provides fields for a two-part address: Virtual Path Identifier (VPI) and Virtual Circuit Identifier (VCI) (see Figure 4). This address combination associates an ATM virtual connection in the physical layer connection. Virtual connections are logical associations between devices in an ATM network. These logical associations can be between a switch and an endstation, between endstations, or between switches in the network. Virtual connections, although requiring physical connections, are less permanent and are created and destroyed by the operations of switches and stations in the ATM network. The physical layer connection may contain one or more virtual paths, and each virtual path may contain one or more virtual circuits. The VPI and VCI addresses are translated within each switch and have local significance only to each switch. That is, each ATM switch charts incoming VPIs and VCIs to outgoing VPIs and VCIs within its switching matrix. Thus addresses are reused throughout the ATM network, as long as each switch takes steps to avoid conflict. There are two main tasks in ATM switching:

- o VPI/VCI translation, and
- o Cell transport from input to its dedicated output port.

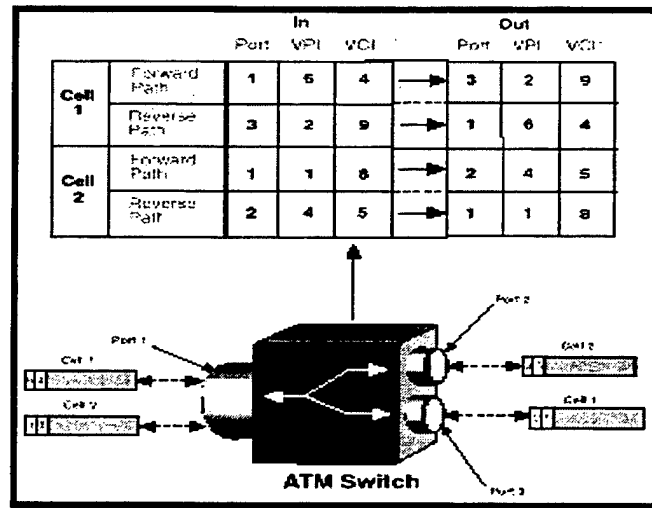
A “switch fabric” provides the physical connection/routing network (see Figure 5) between input and output ports of an ATM switch. A switch table provides the principal reference determining how the cells are processed/routed through the switch. Thus, VPIs and VCIs from input cells in the input ports are translated to the appropriate output ports and the cells are routed appropriately within the switch fabric. A new set of VPIs and

<b>Traffic Class</b>	<b>Timing Relationship</b>	<b>Connection Mode</b>	<b>Bit Rate</b>	<b>Description</b>
Class A	Synchronous	Connection-oriented	Constant Bit Rate (CBR)	Uncompressed voice or video (circuit emulation)
Class B	Synchronous	Connection-oriented	Variable Bit Rate (VBR)	Compressed voice or video (bursty traffic)
Class C	Asynchronous	Connection-oriented	VBR	No timing relationship exist Btwn sender and receiver (TCP/IP, IPX, X.25) and handles data, voice, video. This class sensitive to cell loss but not delay/latency.
Class D	Asynchronous	Connection-oriented	VBR	Switched Multimegabit Data Services (SMDS)

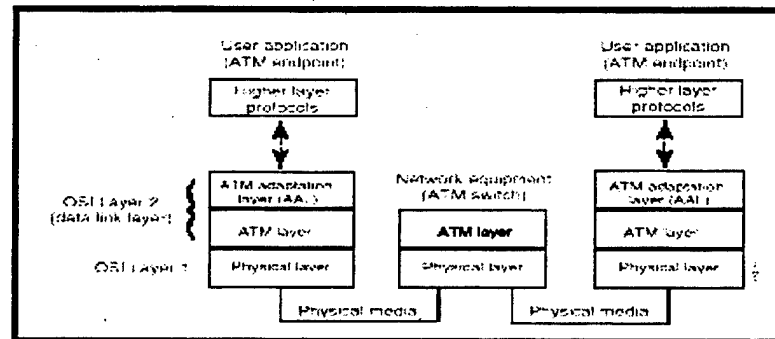
**Table 1.** Traffic classes supported by ATM adaptation layer  
From (Cisco 1996)



**Figure 4.** Virtual path and virtual channel  
From (Cisco 1996)



**Figure 5. Switching table and cell routing**  
From (Cabletron 1997)



**Figure 6. ATM functional elements mapped to OSI layers**  
From (Cisco 1998)

input to the next translation process of the next ATM switch's switch table. Individual Switch tables do not have to know the full path from sender to destination. Each only needs to know to which next switch it must route the arriving cells.

*c. Layers*

ATM functions in only two layers of the OSI model: OSI Layer 1 (Physical layer) and OSI Layer 2 (Data link layer). The physical layer is the interface to the transmission media. Its primary concerns are the physical interface, transmission rates, conversion of cells to line signal and vice versa, physical connector type, clock extraction, and error detection and correction. In the data link layer, the mapping of user information into an ATM format and vice versa occur (see Figure 6). For ATM purposes, there are two sublayers in Layer 2: ATM layer and the ATM adaptation layer (AAL). Table 2 describes the purpose/functions of the ATM layer and the ATM adaptation layer. Further details of the higher layer functions are found in (Wu 1998).

*d. Signaling*

ATM signaling is utilized to dynamically establish, maintain, and terminate ATM connections that are not of the PVC category. It uses permanent or semi-permanent virtual channel connection (VCC) dedicated for sending and receiving signaling messages. That is, the virtual connection used for the connection/call control signaling is not used to send data concurrently ("out-of-band"). This approach allows ATM switches to continually manage/supervise connections without hindering the flow of data. ATM signaling is based on International Telecommunications Union (ITU-T)

Q.2931 protocol, and supports point-to-point and point-to-multipoint switched channel connections.

Higher Layers		Higher layer functions
ATM Adaption Layer	Convergence Sublayer	Provide appropriate traffic Services to higher layer protocols
	Segmentation And Reassembly Sublayer	Segmentation and reassembly
ATM Layer		Generic flow control Cell header generation/extraction Cell VPI/VCI translation Cell multiplexing
Physical Layer	Transmission Convergence Sublayer	Cell rate decoupling HEC sequence generation/verification Cell delineation Transmission frame adaption Transmission frame generation/recovery
	Physical Medium Layer	Bit timing Physical medium

**Table 2.** Functions of ATM layers  
From (Wu 1998)

*e. Technology*

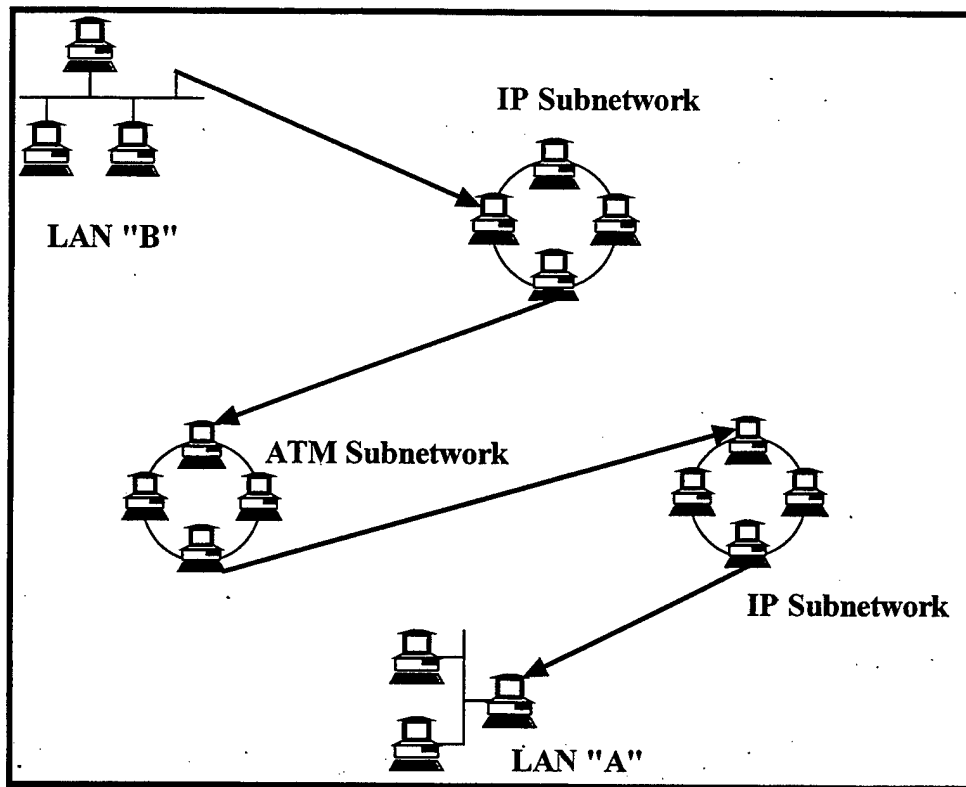
ATM is a transmission, switching and interface technology. It supports both narrowband and broadband speeds (Cisco 1997), and when used as a backbone switch provides the following benefits:

- o Bandwidth efficiency (statistical multiplexing) – bandwidth is shared and only provided , i.e., “on demand”
- o Multiple service support – transports literally any type of information and supports a broad range of user interfaces (SMDS, LAN technology, Frame Relay, etc.)
- o High performance – high bit rate; logical connectivity to various destinations via a single physical interface; dynamic bandwidth sharing
- o Low delay – uses small fixed-sized cell with no error checking and correction of payload at switching layer
- o Flexibility – guarantees minimum amount of bandwidth through several QoS classes
- o Scalable – supports narrowband and legacy interfaces while providing broadband interfaces in anticipation of future growth and emerging ATM services (Cisco 1997).



## 2. Classical Internet Protocol (IP) Over ATM

When IP packets travel within an IP network, both the format and processing of the packets (whether IPv4 or IPv6) are fairly uniform throughout the network (with minor exceptions, as each router manufacturer may add additional processing beyond the requisite protocol processing for marketing purposes). IP packets may be formatted according to their protocol version (IPv4 or IPv6) or maybe in a local area network (LAN) protocol format (e.g., IEEE 802.3/Ethernet, IEEE 802.5/Token Ring). They are variable in length and may grow up to 65K bytes in length for each packet. If the path of an IP packet entails traversing a homogenous ATM network, the packets must be "reconditioned" in order to comply with the data processing architecture of ATM switches. ATM switches (see ATM section II.B.1) process data by cells that are fixed in length and have network characteristics and features that are different than those found in IP networks. At the sending end, ATM Adaptation Layer (AAL) 5 of the ATM processing layers has the responsibility of segmenting user IP packets (heretofore referred to as Protocol Data Unit (PDU)) into 53 bytes *cells* for transport. At the receiving end, AAL 5 reassembles/reconstructs the cells back into the PDU format and forwards the PDU to the upper layer protocols for further processing (see Figure 7). ATM standards assure that on any given ATM virtual connection (VC), cell ordering is maintained end-to-end (Lauback and Halpern 1998). However, it is up to upper layer protocols to determine/request retransmissions of PDUs.



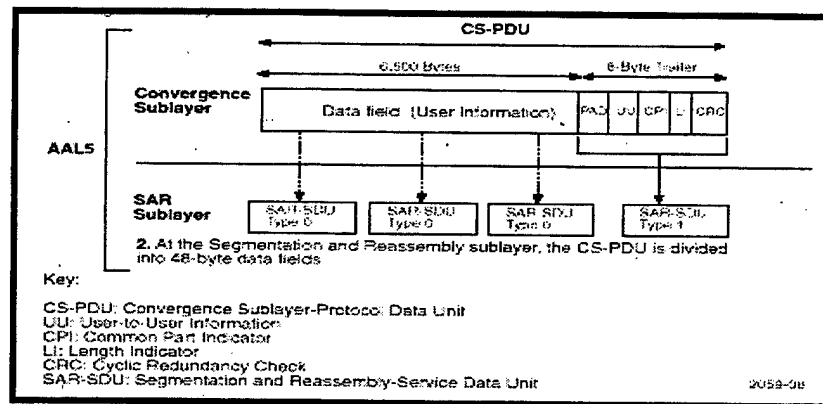
**Figure 7.** System view of classical IP model

According to ATM section **II.B.1**, the ATM Adaptation Layer is divided into two sublayers: Convergence sublayer (CS) and the Segmentation and Reassembly Sublayer (SAR). The CS attaches an IEEE 802.2 Logical Link Control (LLC) header and an IEEE 802.1A SubNetwork Attachment Point (SNAP) header to encapsulate the PDUs and separate each according to senders and destinations (see Figure 8). The LLC/SNAP encapsulation allows ATM to multiplex PDUs to one VC instead of dedicating a VC for each protocol. The PDU (up to a maximum length of 65000 bytes) along with the LLC/SNAP header is attached to a trailer consisting of a pad, user-to-user (UU) information, common part indicator (CPI), length indicator (LI), and a cyclic redundancy check (CRC). This new, whole assembly forms a new CS-PDU. The SAR accepts the CS-PDU from the convergence sublayer and segments it into 48-byte data fields called SAR Sublayer-Service Data Unit (SDU). Each SAR-SDU is labeled a type "0" until the last 48-byte segment which is labeled type "1". At the ATM layer, each SAR or SDU is appended with a 5-byte header, thus forming the ATM 53-byte cells. The payload type (PT) field in the cell header is set to type "0", and the last cell of the PDU carries the type "1".

At the destination, the ATM layer extracts the PDU data field from each cell and sends it to the SAR sublayer for reassembly into CS-PDU. To verify correct transmission and reassembly, the CS checks the CRC and length field. After which, the reconstituted PDU is forwarded to higher layers of the ATM model for further processing.

The process of receiving the variable-length PDU and processing it through the AAL and the ATM layer at both the entry and exit points of the ATM network injects

latency into the end-to-end (sender-to-destination) process. The routing/switching within the ATM network—once the PDU has been processed into 53-byte cells—maintains the high-speed switching, high throughput characteristics of ATM. As with the case of the overall ATM model, related security issues are not considered in transporting IP packets through an ATM subnetwork (Lauback and Halpern 1998).



**Figure 8. Cell preparation**  
 From (Cabletron 1997)

## D. OSI LAYER 3 (NETWORK LAYER) PROPOSALS

### 1. Security Architecture for IP (IPsec) Protocol

The Security Architecture for the Internet Protocol (IPsec) proposes security services in the IP layer for both the IPv4 and IPv6 environments, using a combination of cryptographic and protocol security mechanisms. These services are intended to render interoperable, high quality, cryptographically-based security, and include the following: Access control, connectionless integrity, data origin authentication, protection against replays (partial sequence integrity), confidentiality (encryption), and limited traffic flow confidentiality (Kent and Atkinson 1998). IPsec focuses on two traffic security protocols

and the cryptographic key management procedures and protocols to achieve its goals. The two security protocols are the Authentication Header (AH) and the Encapsulating Security Payload (ESP). The recommended key management protocol for IPsec is the Internet Key Exchange.(Harkins and Carrel 1998) Although IPsec affords protection within the IP layer, it also provides similar security services to upper layer protocols (TCP, UDP, ICMP, BGP, etc.) at the same time. The protection mechanisms of IPsec are intended to be cryptographic algorithm-independent. That is, selection of algorithms will not affect other sections of implementation. The algorithm choice will definitely be a factor in overall security.

*a. Security Policy Database (SPD)*

In IPsec, the extent/level of protection of each IP packet is determined either by the application layer or requirements defined in a Security Policy Database. The SPD characterizes the local security policy and is maintained by the user or a system administrator in a host or a security gateway environment (router or a firewall). Outbound packets, whether branded by the application layer or not, are matched against the SPD to determine the type of processing to be applied: IPsec security furnished, packet discarded, or IPsec bypassed. Control of security, key management and traffic flow are the purview of the SPD. When security services are required for an IP packet, IPsec, with assistance from the SPD, assigns the required security protocol and cryptographic algorithm, and puts in place the necessary cryptographic keys. Protection is provided to one or more connectivities between two hosts, two security gateways, or between a host and a security gateway.

**b.     *Operation***

The user determines the granularity of the security services applied to each IP packet through the application layer in conjunction with the SPD. As data is processed through the protocol stack, IPsec determines the level of protection/processing the data requires in accordance with the SPD. If the data requires connectionless integrity, data origin authentication, and/or anti-replay services, then the Authentication Header (AH) security protocol is applied. If the data requires confidentiality/traffic flow confidentiality, then the Encapsulation Security Payload (ESP) is applied. The AH and ESP may be applied alone or in combination with each other. Furthermore, AH and ESP are both capable of operating in two modes: transport mode and tunnel mode.

Note: A more detailed explanation of the AH and ESP security protocols is provided in section II.C.2 and II.C.3 respectively.

**c.     *Key Management***

Shared secret value values (cryptographic keys) are required to implement the security services of AH and ESP. Key management/distribution (automatic or manual) for cryptographic algorithms utilized are handled by mechanisms separate from IPsec. IPsec recommends the Internet Key Exchange (IKE) protocol, but other key distribution techniques such as Kerberos and SKIP may be employed.

In order to identify the extent/level of security protection attributed to each packet, IPsec uses the concept of a "*Security Association*" (SA). As defined in its Internet draft submission:

A Security Association is a simplex "connection" that affords services to the traffic carried by it. Security services are afforded to a SA by the use of AH, or ESP, but not both. If both AH and ESP protection is applied to a traffic stream, then two (or more) SAs are created to afford protection to the traffic stream. To secure typical, bi-directional communication between hosts, or between two security gateways, two Security Associations (one in each direction) are required. (Kent and Atkinson 1998)

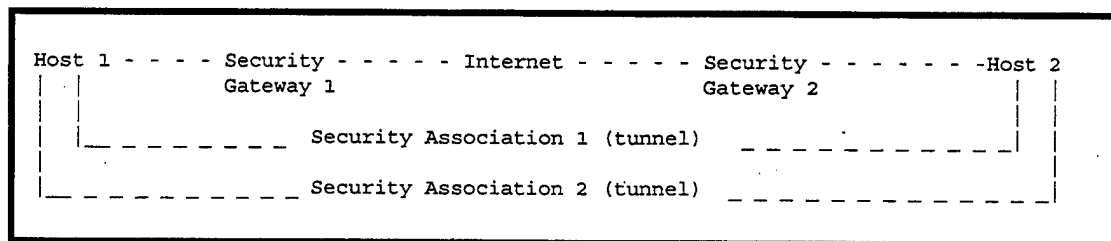
Each SA is negotiated and agreed upon by all parties involved before actual data transfer occurs. To uniquely identify a SA, IPsec uses a combination of an IP destination address, a security protocol identifier (AH or ESP), and a Security Parameter Index (SPI). SPI is a 32-bit value used to differentiate SAs having the same IPsec protocol and ending at the same destination. As with AH and ESP, there are two types of SAs defined: transport mode and tunnel mode.

A transport mode SA only exists between two or more hosts. In the transport mode, the security protocol header is inserted between the IP header (including any options) and upper layer protocols for IPv4, and between the base IP header (including extensions, but before or after destination options) and upper layer protocols in IPv6. Transport mode SAs afford security services only to upper layer protocols in the case of ESP, while security services are extended to selected portions of the IP header in the case of AH. (See the AH section for more details.)

A tunnel mode SA exists between two or more hosts and between two or more security gateways. Any instance of a SA between security gateways must be in tunnel mode, except for the case where the security gateway is receiving traffic as a host (e.g., simple network management protocol (SNMP) messages). An "outer" IP header is used in the tunnel mode, and the IPsec security protocol header is placed between the

outer IP header and the "inner" IP header. Similar to the transport mode, tunnel mode SAs afford security services only to upper layer protocols in the case of ESP, while security services are extended to selected portions of the IP header in the case of AH.

On the occasion that a combination of AH and ESP security associations is required, a SA "bundle" is used. The order of the sequence in the bundle is defined by the mandating security policy. Also, termination of a bundled SA may occur at different endpoints as illustrated in Figure 9.



**Figure 9.** Combining of security associations  
From (Kent and Atkinson 1998)



Basic combinations of SAs can be one of four cases (Figures 10-13):

Note:

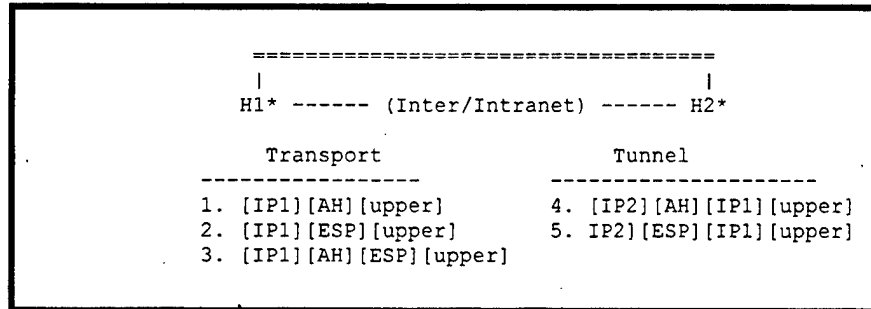
===== SAs (AH or ESP, transport or tunnel)

----- = connectivity

SGx = security gateway

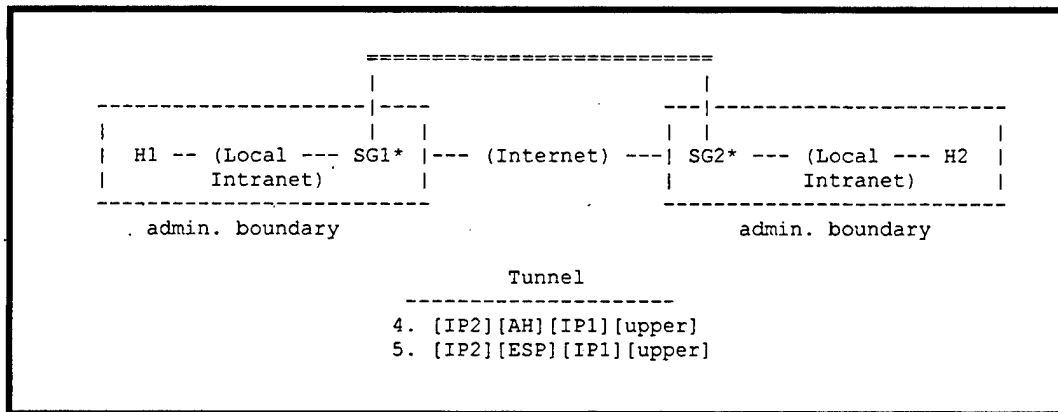
X\* = X supports IPsec

#### Case 1. End-to-end security between 2 hosts



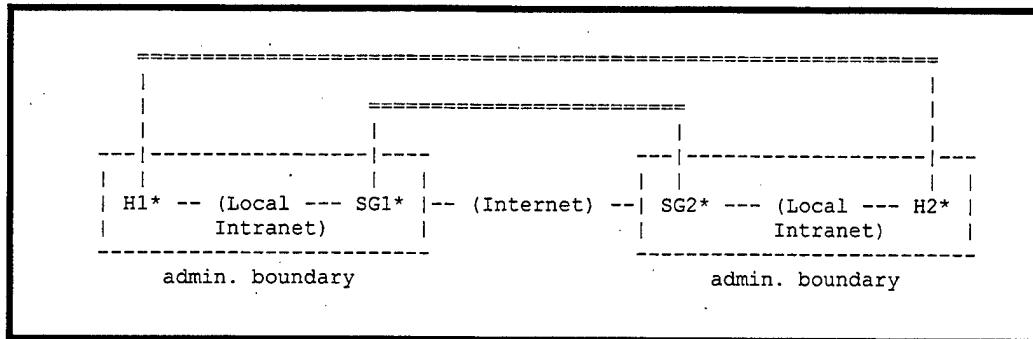
**Figure 10.** End-to-end security between 2 hosts  
From (Kent and Atkinson 1998)

#### Case 2. Simple virtual private network (VPN)



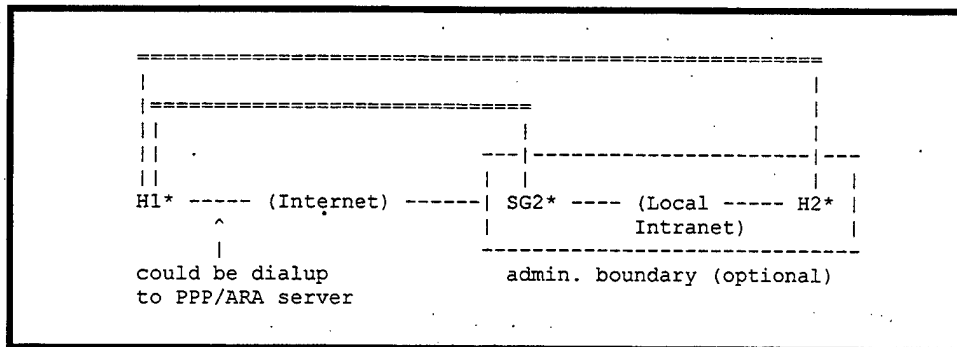
**Figure 11.** Simple VPN  
From (Kent and Atkinson 1998)

Case 3. Combination of case 1 and 2



**Figure 12.** Combination of end-to-end and VPN  
From (Kent and Atkinson 1998)

Case 4. Remote/mobile (e.g. dial-up) host and VPN

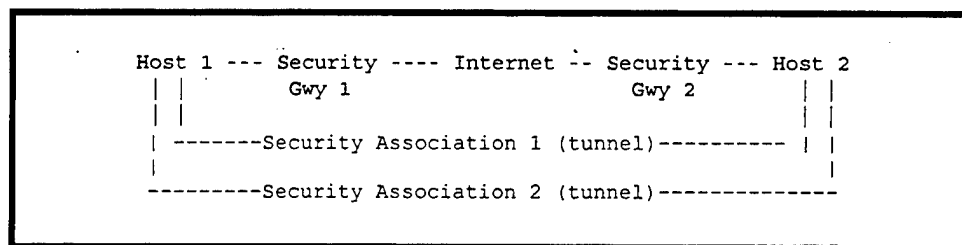


**Figure 13.** Remote/Mobile host and VPN  
From (Kent and Atkinson 1998)

Security Associations can be joined into a bundle in two ways: transport adjacency and iterated tunneling. Transport adjacency occurs when the AH and ESP protocols are combined and applied to the same IP datagram without invoking tunneling

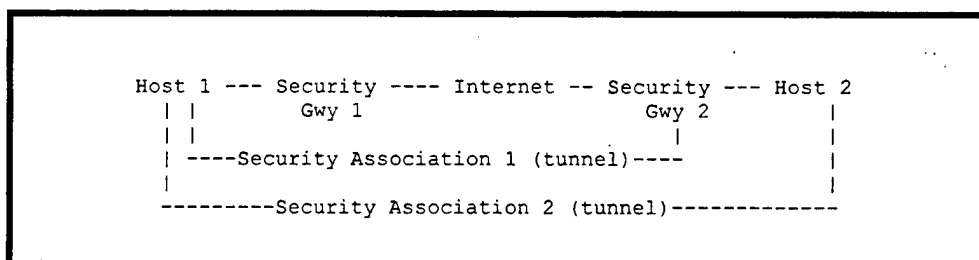
(see Figure 13). Iterated tunneling is the layering of multiple security protocols through tunneling. The tunnels themselves can be further nested since each tunnel can originate and terminate at various IPsec processing along the IP path. IPsec accomplishes iterated tunneling in three basic ways:

1. Endpoints for SAs are the same; inner and outer tunnels can either be AH or ESP



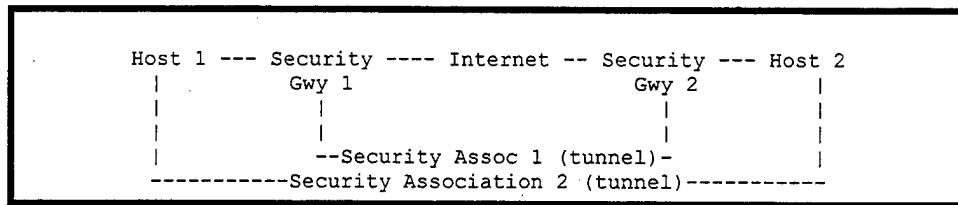
**Figure 14. Same SA endpoints**  
From (Kent and Atkinson 1998)

2. Only one endpoint is the same; inner and outer tunnels can either be AH or ESP.



**Figure 15. One endpoint is the same**  
From (Kent and Atkinson 1998)

3. Neither endpoint is the same; inner and outer tunnels can either be AH or ESP.



**Figure 16. End points are different**  
From (Kent and Atkinson 1998)

**d. Security Association Database (SAD)**

As one of the two nominal databases in the IPsec model, the SAD manages the security parameters assigned to each security association. All SAs (single or bundled) created for each session have an entry in the SAD. Entries in the SAD are indexed by the destination address, IPsec protocol type and the Security Parameter Index (SPI). During outbound processing of IP packets, the SPD points to the SAD for entries of security parameters for existing SA session. If no existing SAD entries equate, a new entry is created. For inbound packets, the SAD is consulted directly to determine the processing required. SA (or SA bundling) granularity (fine-grained or coarse grained) is dependent on the outcome of the initial set up negotiation and/or on local security policy. Possible scenarios include:

- o A single SA with a uniform set of security services, for all traffic between two or more hosts.

- o Security services distributed among a number of SAs, for all traffic between two or more hosts.

The same set of alternatives maybe applied by two security gateways.

*e. Performance Issues.*

The focus of IPsec protocols is primarily security. Therefore implementation of the IPsec protocols do impose computational and memory costs on hosts or security gateways. These costs are:

- o Required memory for IPsec code and data structures
- o Computation of integrity check values
- o Per packet encryption and decryption
- o Software-based cryptography
- o Bandwidth utilization on transmission, switching, and routing, caused by components not implementing IPsec and the increased bit overhead due to the use of AH and ESP
- o Increased packet traffic associated with key management

These per-packet costs are manifested in increased latency and reduced throughput (Kent and Atkinson 1998).

## 2. Authentication Header Protocol

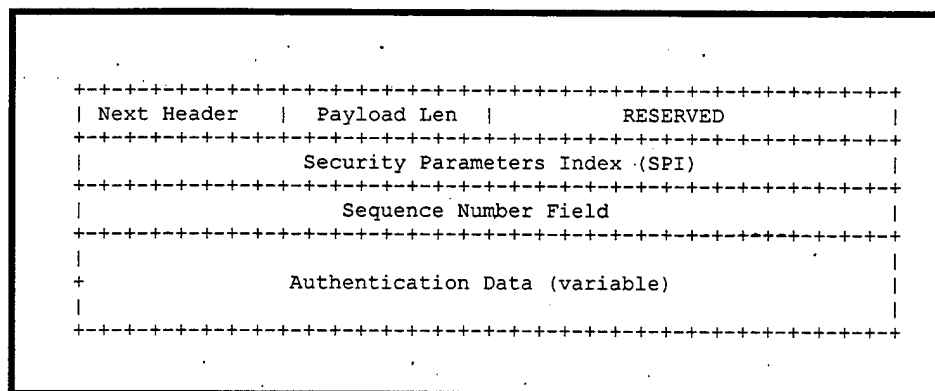
As one of two traffic security protocols of the IP Security Architecture, the IP Authentication Header (AH) Protocol provides security services to IP packets in two major areas:

- o Data origin authentication and connectionless integrity for IP datagrams
- o Protection against replay (Kent and Atkinson 1998)

In addition to IP headers, AH also provides authentication to upper layer protocol fields as well. It is known that IP header fields may change in transit. A number of these fields will have final values that are unpredictable upon arrival at destination.

### a. Format

The AH header format is compatible with both IPv4 and IPv6. It consists of a mandatory 12 bytes of fixed length fields plus a variable-length field. The total length is an integral multiple of four bytes in length.



**Figure 17. AH format**  
From (Kent and Atkinson 1998)

These fields are:

- o Next Header - an 8-bit field identifying type of payload after AH
- o Payload Length - an 8-bit field specifying the length of AH in 32-bits words (4-byte units).
- o Reserved - a 16-bit field reserved for future use. The reserved field is set to "zero" and is a part of the authentication data calculation.
- o Security Parameter Index (SPI) - A 32-bit arbitrary value which uniquely identifies the Security Association for a datagram (see section II.C.3).
- o Sequence Number - a 32-bit field, which contain an increasing/decreasing counter value.
- o Authentication Data - a variable length that holds the Integrity Check Value (ICV) for the IP packet. The authentication data field must be an integral multiple of 32 bits in length. (Reynolds and Postel 1994)

**b. Operations**

(1) Integrity Check Value (ICV) Calculation. The computation of the ICV is performed over the following elements of the IP packet:

- o IP header fields that are immutable in transit or predictable in value upon arrival at its destination
- o AH header (Next header, payload length, reserved, SPI, sequence number, and authentication data (set to zero initially during computation) and any explicit padding bytes
- o Upper level protocol data (assumed to be immutable during transit)

For mutable fields, AH sets the value to zero for computational purposes. When a field is mutable but its final value at the destination is predictable, then the predicted final value is utilized for the computation of the ICV (see Table 3). If AH encounters an extension header that it does not recognize, it will discard that IP packet and transmit an ICMP message to the origin. The table below describes the division of fields for IPv4 and IPv6 into immutable, mutable-but-predictable, and mutable (zeroed prior to ICV computation):

The SA to which the SPD is pointing determines authentication algorithm applied in the ICV computation. The AH protocol supports a good number of cryptographic algorithms, and the use of a specific algorithm will be determined by local security policy. As an example, keyed Message Authentication Code (MAC) based on a symmetric encryption algorithm (e.g., Digital Encryption System (DES)) or a one-way hash functions (e.g., Secure Hash Algorithm-1 (SHA-1)) is a suitable authentication algorithm for point-to-point connectivity. One-way hash algorithms combined with asymmetric signature algorithms are appropriate for multicast connectivity. However, to

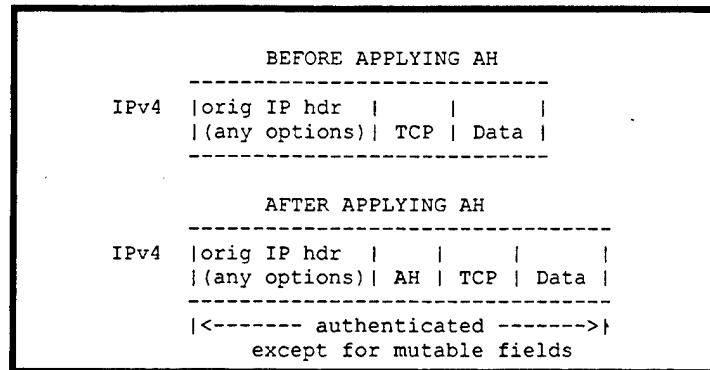


achieve minimum AH implementation compliance, an AH implementation must support the use of HMAC (Key-Hashing for MAC) with MD5 (Madson and Glenn, 1998) and HMAC with SHA-1 (Madson and Glenn, 1998).

Version	Immutable	Mutable/ Predictable	Mutable
IPv4	<ul style="list-style-type: none"> <li>- Version</li> <li>- Internet header length</li> <li>- Total length</li> <li>- Identification</li> <li>- Protocol</li> <li>- Source address</li> <li>- Destination address</li> </ul>	<ul style="list-style-type: none"> <li>- Destination address</li> </ul>	<ul style="list-style-type: none"> <li>- Type of service (TOS)</li> <li>- Flags</li> <li>- Fragment offset</li> <li>- Time-to-live (TTL)</li> <li>- Header checksum</li> </ul>
----- Optional Fields	<ul style="list-style-type: none"> <li>- End of option list</li> <li>- No operation</li> <li>- Security</li> <li>- Extended security</li> <li>- Commercial security</li> <li>- Router alert</li> <li>- Sender directed multi-destination delivery</li> </ul>	-----	<ul style="list-style-type: none"> <li>- Loose source route</li> <li>- time stamp</li> <li>- record route</li> <li>- strict source route</li> <li>- traceroute</li> </ul>
IPv6	<ul style="list-style-type: none"> <li>- Version</li> <li>- Payload length</li> <li>- Next header</li> <li>- Source address</li> <li>- Destination address</li> </ul>	<ul style="list-style-type: none"> <li>- Destination address</li> </ul>	<ul style="list-style-type: none"> <li>- Class</li> <li>- Flow label</li> <li>- Hop limit</li> </ul>
----- Optional fields	-----	<ul style="list-style-type: none"> <li>- Routing</li> </ul>	<ul style="list-style-type: none"> <li>- Hop by hop options</li> <li>- Destination options</li> </ul>

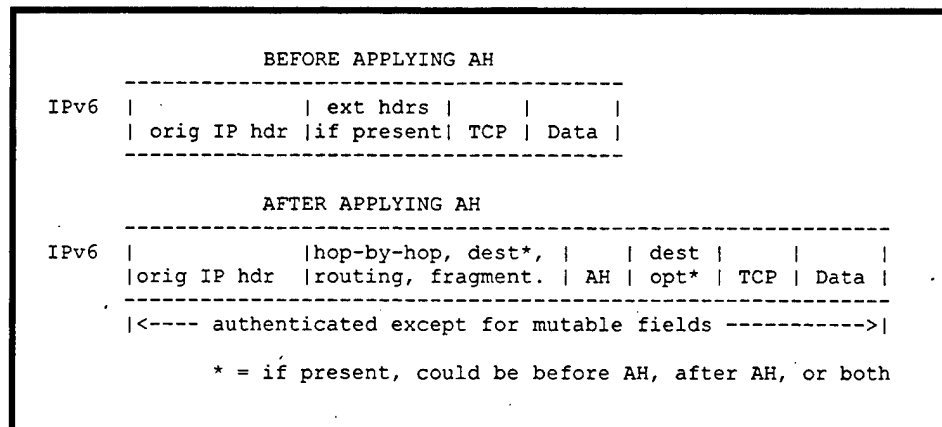
**Table 3. IPv4 & IPv6 Mutable/Immutable fields**  
After (Kent and Atkinson 1998)

(1) Processing. Employment of AH is accomplished in two ways: transport mode or tunnel mode. In the transport mode, AH is inserted after the IP header and before an upper layer protocol (e.g., TCP, UDP, ICMP, etc.) (see Figure 18 and 19).



**Figure 18. IPv4 before & after AH applied**  
From (Kent and Atkinson 1998)

In this fashion, security service is provided mostly to the upper layer protocols, but the ICV computation does include the original IP header plus subsequent immutable fields/headers, which appears before the upper layer, protocols. In the tunnel mode, AH provides security services to the entire inner IP packet, including the entire IP header. The outer IP header is included in the computation of the ICV with the exception of

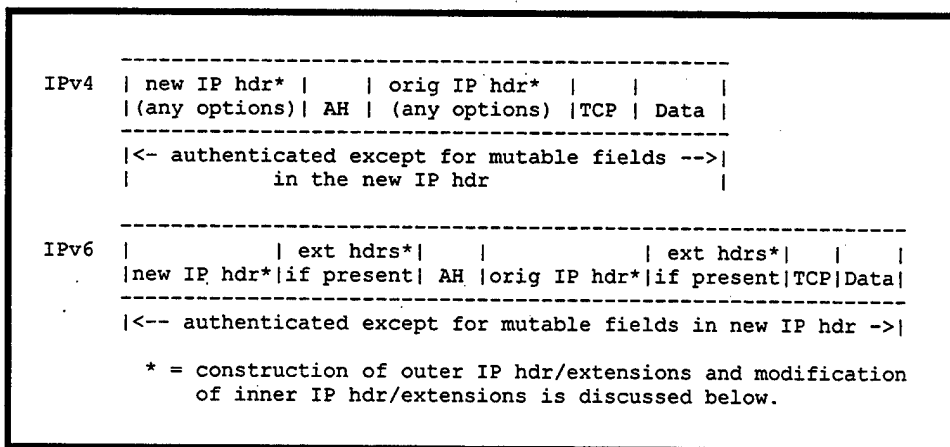


**Figure 19. IPv6 before & after AH applied**  
From (Kent and Atkinson 1998)

mutable fields/headers. Tunnel mode is employable in either hosts or security gateways. However, when implementing AH in security gateways, only the tunnel mode can be

used. The inner IP header contains the final source and destination addresses, while the outer IP header may carry distinct IP addresses such as those of security gateways.

Application of AH to any outbound packet is conducted only after an IPsec implementation has determined that the outbound packet is matched to a SA that calls for AH processing. In the inbound cycle, the receiving host ascertains the level of IPsec processing (AH, ESP or both) by examining the destination address, security protocol, and SPI (see IPsec section). If the IP packet does not equate to a valid SA association, the packet is discarded and an audit log entry is made. The protocol requires that all AH implementation support anti-replay service through the use of a sequence number. At the outset of the communication, the sender establishes a counter for each SA and sets



**Figure 20. IPv4 & IPv6 authenticated fields**  
From (Kent and Atkinson, 1998)

its value to 0. This sequence number is incremented for each packet sent for each associated SA until it cycles back to zero. Any packet found carrying a duplicate sequence number is discarded and an audit log entry made. The anti-replay feature is

enabled by default by the sender unless otherwise instructed by the receiver. In the event the receiver disables that anti-replay, the sender still increments the counter, but need not monitor nor reset it.

Padding is applied to the AH header to ensure the length is a multiple of 32-bits (IPv4) or 64 bits (IPv6). The amount of padding is determined by the length of the ICV (further determined by the algorithm used) and the IP protocol version. The value zero is used for the padding octets.

Enroute to the destination, AH-protected packets are subjected to possible fragmentation due to dynamic conditions that change the maximum transmission unit (MTU) for the path and links. If needed, reassembly is performed before AH processing occurs. However, AH processing on a packet is only permitted on those packets with their OFFSET field value set to zero or MORE FRAGMENT flag not set upon arrival. In other words, if a packet, appearing to be an IP fragment, arrives with a non-zero value in its OFFSET field and/or the MORE FRAGMENTS flag set, the packet is discarded and an audit log entry made.

### **3. Encapsulating Security Payload (ESP) Protocol**

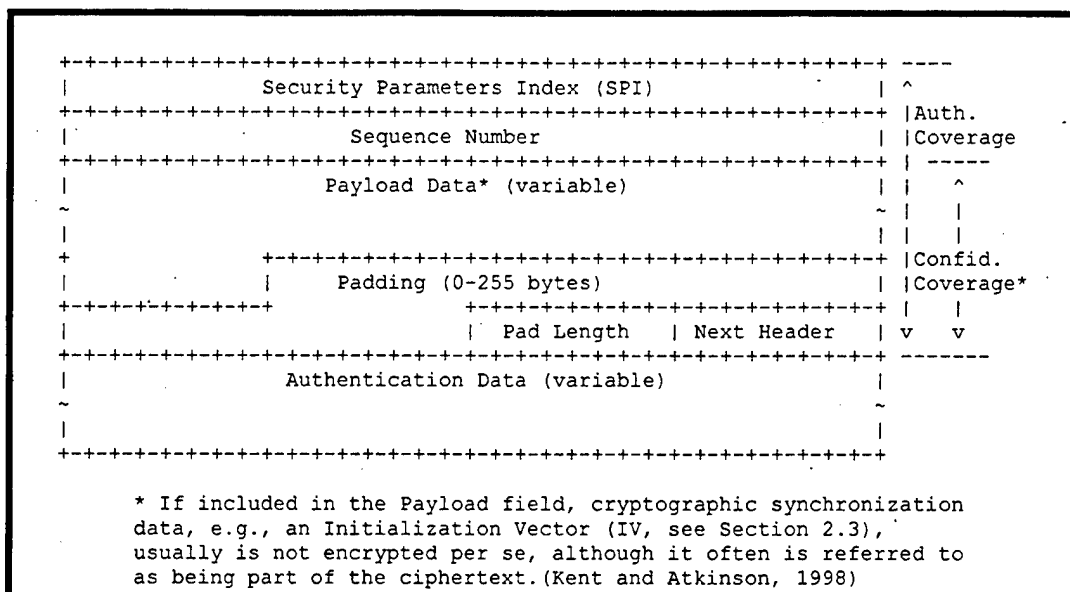
The second of the two traffic security protocols in the IP Security Architecture, ESP provides security services to IP packets by means of data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and moderate traffic flow confidentiality. ESP achieves confidentiality by encrypting the data portion of the IP packet. The local security policy may dictate whether encryption is confined to the transport-layer segment (e.g., TCP, UDP, ICMP, IGMP, etc.) or the entire

IP datagram. Data authentication and connectionless integrity are achieved through the use of an authentication data field, much like AH. The anti-replay service is also identical to AH. By default this field includes a Sequence Number parameter which the sender increments for each packet. Checking this parameter to utilize the anti-replay security service is at the discretion of the receiver. Data origin authentication is solely at the discretion of the receiver. The default calls for the sender to increment the Sequence Number used for anti-replay regardless of the receiver's action on the Sequence Number at its end.

ESP has two components: The unencrypted and the encrypted payload. The unencrypted field(s) of the ESP header inform the destination how to properly decrypt and process the encrypted data. The encrypted field(s) are protected fields for the security protocol and the encapsulated IP datagram. ESP is applicable to IPv4 and IPv6 and operates in two modes: Transport and tunnel mode.

**a. Syntax**

The ESP header (see Figure 21) is inserted after the IP header and before the upper layer protocol header (transport mode) or before an encapsulated IP header (tunnel mode).



**Figure 21. ESP header elements**  
From (Kent and Atkinson 1998)

- o Security Parameters Index - A 32-bit arbitrary value which uniquely identifies the Security Association for a datagram
- o Sequence Number - a 32-bit field, which contain an increasing/decreasing counter value
- o Payload Data – a mandatory, variable-length (integral number of bytes in length) field which contains data described by the Next Header field, and may contain cryptographic synchronization data such as an Initialization Vector (IV), encryption algorithm, or per-packet synchronization data (length, structure, location).

- o Padding (for Encryption) – a variable-length field with a maximum length of 255 bytes. Utilized under a number of packet conditions:

- o If an encryption algorithm is employed that requires the plaintext to be a multiple of some number of bytes, e.g., the block size of a block cipher, the Padding field is used to fill the plaintext (consisting of the Payload Data, Pad Length and Next Header fields, as well as the Padding) to the size required by the algorithm.

- o Padding also may be required, irrespective of encryption algorithm requirements, to ensure that the resulting ciphertext terminates on a 4-byte boundary. Specifically, the Pad Length and Next Header fields must be right aligned within a 4-byte word, as illustrated in the ESP packet format figure above, to ensure that the Authentication Data field (if present) is aligned on a 4-byte boundary.

- o Padding beyond that required for the algorithm or alignment reasons cited above, may be used to conceal the actual length of the payload, in support of (partial) traffic flow confidentiality. However, inclusion of such additional padding has adverse bandwidth implications and thus its use should be undertaken with care. (Kent and Atkinson 1998).

- o Pad Length – a mandatory field that indicates the number of pad bytes immediately preceding it. The range of valid values is 0-255, where a value of zero indicates that no Padding bytes are present.



- o *Next Header* – a mandatory 8-bit field that identifies the type of data contained in the Payload Data field. The value of this field is chosen from the set of IP Protocol Numbers defined in the "Assigned Numbers" [STD-2] RFC from the Internet Assigned Numbers Authority (IANA)
- o *Authentication Data* – an optional field, it is variable-length containing an Integrity Check Value (ICV) computed over the ESP packet minus the Authentication Data. The selected authentication algorithm determines the length of the field, rules and processing steps for validation. The SA AH in conjunction with ESP will mandate the selection of the authentication field over the use of in question.

***b. Algorithms***

The SA in question specifies the encryption algorithm employed in the ESP. Symmetric encryption algorithms are more suitable to ESP. Much like the AH, the authentication algorithm for the authentication field is also specified by the SA in question and the selection of algorithms is the same as those available for the AH.

***c. Transport Mode***

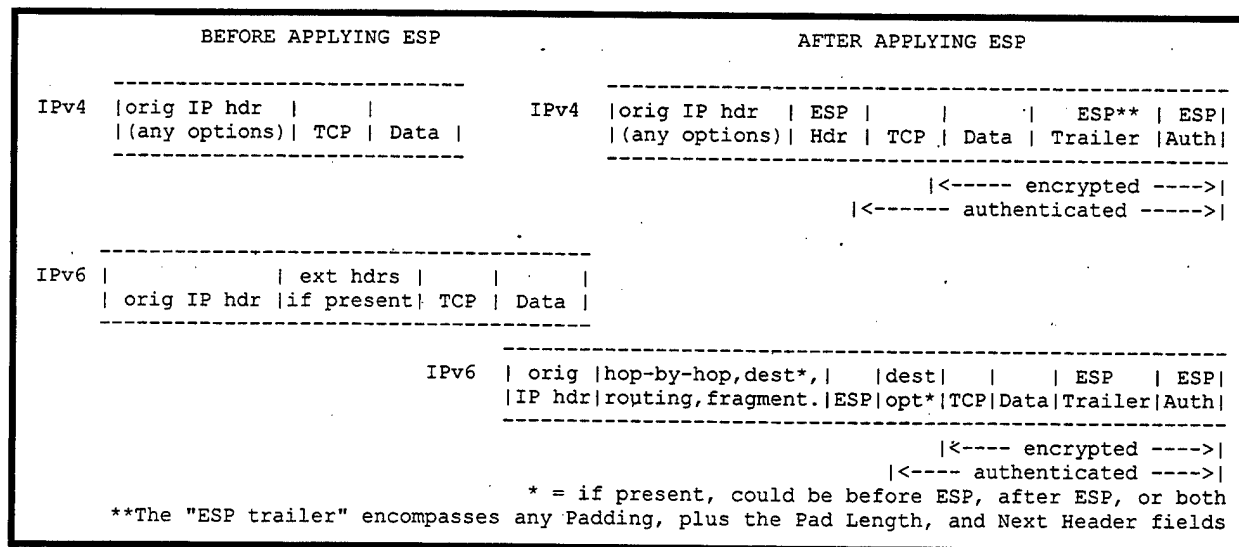
The transport mode is applicable only to host implementations, providing security services to upper layer protocols only and not the IP header. The ESP header is inserted after the IP header and before any upper layer protocols (e.g., TCP, UDP, ICMP, etc.) or any other IPsec headers (e.g., AH) already inserted. For IPv4, the ESP header is placed after the IP header (and any IP header options) and before upper layer protocols.

For IPv6, the ESP header appears after the hop-by-hop, routing, and fragmentation extension headers (see Figure 22). For outbound packets, the sender encapsulates upper layer protocol information into the ESP header/trailer.

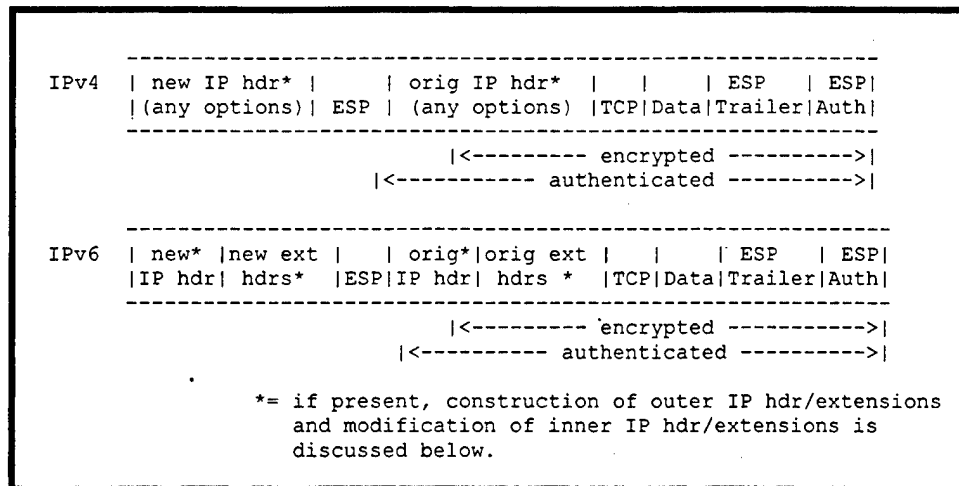
**d. Tunnel Mode**

ESP in tunnel mode is employable in either hosts or security gateways.

When implementing in security gateways, ESP must be in tunnel mode. The packet structure follows that of packet tunneling principle (see Figure 23): There is an "inner" IP header that carries the addresses of the ultimate source and destination, and an "outer" IP header which carries a distinct IP address, that of a security gateway. In the tunnel mode, the entire inner IP packet is provided the security protection, including the entire inner IP header. The positioning of the ESP header in tunnel mode is the same as in the transport mode.



**Figure 22.** ESP header transport mode application to IP packet  
From (Kent and Atkinson 1998)



**Figure 23.** ESP header tunnel mode application to IPpacket  
From (Kent and Atkinson 1998)

#### *e. Outbound Processing*

The following are the sequence of events occurring when ESP security services are applied in processing outbound IP packets:

1. Encapsulation of upper layer protocol information into the ESP Payloadfield:
  - a) For transport mode – the original upper layer protocol information
  - b) For tunnel mode – the entire original IP datagram
2. Addition, if any, of necessary padding.
3. Encryption of resulting bitstream from ESP trailer up to and including the following fields of the ESP header: Payload data, padding, pad length, and next header.
4. When the option of authentication is selected, encryption is conducted first before applying the authentication algorithm. The authentication field itself is

not encrypted. A keyed authentication algorithm must be used to compute the ICV since the resulting authentication data is not protected by the encryption within ESP.

*f. Inbound Processing*

The following are the sequence of events occurring when ESP security services are applied in processing inbound IP packets:

1. The arriving IP packet is reassembled if fragmentation occurred during transit. If the OFFSET field is non-zero or the MORE FRAGMENTS flag is set, the packet is discarded and an audit log entry made (see IPsec policy on fragmented IP packets).
2. Based on destination IP address, security protocol (ESP), and the SPI, the receiver determines the applicable SA. Using the SA, the inbound processing will check Sequence Number, if the feature is turned on, and specify the algorithms for decryption and keys for generation of the ICV.
3. Using the key, encryption algorithm, algorithm mode, and cryptographic synchronization data (if any), the ESP Payload Data, Padding, Pad Length, and Next Header fields are decrypted. If the Authentication field is present, the ICV is computed and the IP packet authenticated.

4. The upper layer protocol information is decrypted using the key, encryption algorithm, and algorithm mode specified in the SA.
5. The original IP datagram is reconstructed

#### **4. Flow Based Security Protocol**

##### **a. Description**

Suvo Mittra (Stanford University) and Thomas Woo (Bell Laboratories) have proposed a security protocol that exploits IP datagram service's advantages (simplicity, flexibility, robustness, and scalability(Mittra and Woo, 1997)), and uses the notion of "flow" as the basis for secure communication. The Flow-Based datagram Security (FBS) protocol uses zero-message keying to preserve connectionless nature of the datagram service (hereupon referred to as datagram semantics), while using soft state to provide per-packet processing efficiency similar to that of a session-oriented scheme.

##### **b. Datagram semantics**

Datagram semantics are attributes describing those of independent IP packets, each independently transmitted, routed and received. Datagram semantics maintain that setup procedures between sender and destination are not required, and neither does an active state exist for the duration of a session. In other words, datagram semantics reflect the core properties of a connectionless-based model. Networking protocols such as Internet Protocol (IP), Unit Data Protocol (UDP) and Remote Procedures Call (RPC) have underlying datagram semantics core.

*c. Flow*

According to the IPv6 specification, a flow is a "sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers." The special handling is often associated with Class-of-Service (CoS) treatment of packets. In general, a flow is used to refer to a set of packets that should be treated uniformly (by the network or the host devices) for better performance. An example of flow might be a bit stream composed of video, audio and text data. Each type are treated as individual flows and accorded the appropriate CoS as it traverses the network. Therefore packets containing video data are dealt with in the same manner at each processing node (routers) according to the assigned CoS to video. Datagrams from an application-to-application "conversation" constitute a flow, and datagrams in a TCP session (OSI layer 4) also constitute a flow. Datagrams of a particular flow receive similar treatment from network nodes throughout their travel, thus logically inheriting characteristics of a connection. Subsequently, a flow exhibits the flavor of both datagram and a connection.

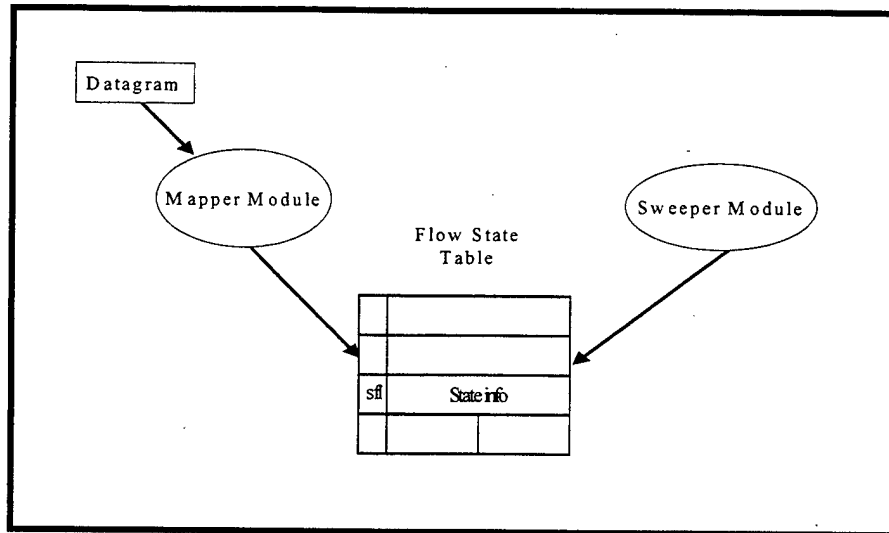
*d. Protocol Overview*

The FBS protocol consists of two mechanisms: A flow-association mechanism (FAM) and a zero-keying mechanism (ZKM). FAM isolates and differentiates datagrams to create flows according to application requirements. ZKM establishes the security parameters for a flow without contracting an end-to-end set up exchange. Working together, the output of FAM feeds into ZKM, which then produces the per-flow cryptographic session key.

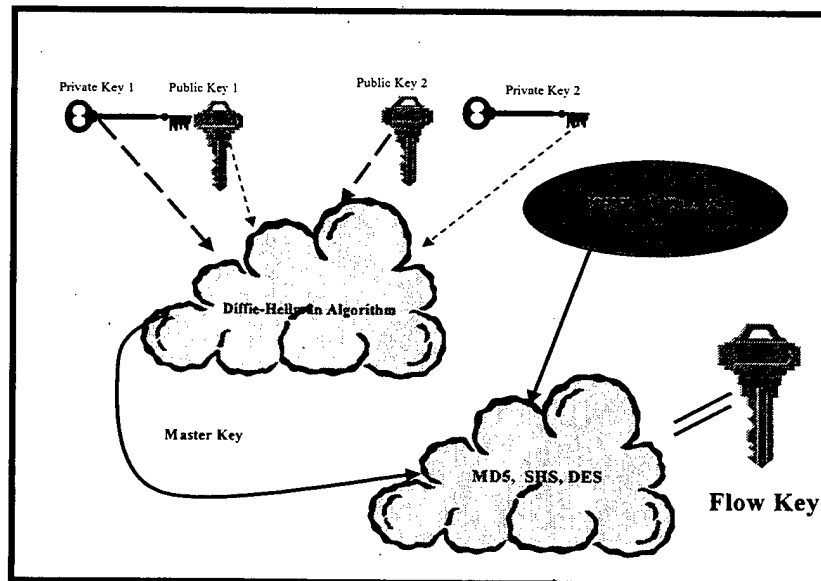
The FBS protocol desires that the FAM should be independent of the security policies. Thus policies are expressed in a policy module (mapper and sweeper modules), which in turn is connected to the FAM to provide guidance in the manner of security processing of the flow. The FAM design is composed of three key elements:

- o Flow state table - contains information of each active flow and the necessary state data required to assist in the operation of the mapper and sweeper modules below.
- o Mapper module - takes datagram attributes (source & destination address, process ID, time, etc) and produces an index, which uniquely identifies the flow. This index is referred to as the *security flow label (SFL)* and is recorded in the flow state table. A new sfl is recorded for each new flow. (see Figure 24)
- o Sweeper module - monitors the flow state table and removes expiring flows from the flow state table.

(1) Zero-based keying. The zero-based keying mechanism uses the basic Diffie- Hellman key exchange model. A pair-based master key is derived from the application of the Diffie-Hellman algorithm. The sfl is then concatenated with the derived master key, and the result fed into a one-way cryptographic hash function (e.g., Message Digest (MD) 5, Secure Hash Standard (SHS), and Digital Encryption Standard (DES)). The outcome of the hashing operation is the cryptographic flow key (see Figure 25).



**Figure 24.** Flow state table  
After (Mittra and Woo 1997)



**Figure 25.** Zero-based keying mechanism



(2) Security Flow Header. A security flow header (FBS header) (see Figure 26) is created and attached to each datagram packet in order to maintain the datagram semantics. The FBS header can be placed between the IP header and the payload (a form of encapsulation), or can be made as an optional section of the IP header itself. The FBS header construction is described below.

Security Flow Label (sfl)	Confounder	Message Authentication Code (MAC)	Timestamp
------------------------------	------------	---	-----------

**Figure 26.** FBS header  
From (Mittra and Woo 1997)

- o *Confounder* - statistically random value generated on per datagram basis. Initialization vector (IV) for encryption. Also used for computation of the MAC. It is used to hide presence of identical datagrams in flow.
- o *Message authentication code (MAC)* - keyed on flow key and calculated over confounder, timestamp and payload:

$$\text{MAC} = (\text{HMAC}(\text{FlowKey}|\text{Confounder}|\text{timestamp}|\text{payload}),$$

where HMAC is a keyed one way hash function.

Ensures integrity of datagram body and other fields in security header. Provides form of flow authentication (i.e., datagram belongs to flow indicated)

- o *Timestamp* - time value for countering replay attacks.

*e. Protocol Operation (see Figure 27)*

(1) Sender. As a stream of datagrams arrives at the FAM of the sender, they are inspected and classified into flows. The appropriate public and private keys, derived from a previous execution of the Diffie-Hellman key exchange protocol, are applied to obtain the master key. The resulting SFL from the FAM is concatenated with the master key and fed into a one-way cryptographic hash function to produce the session flow key (see Figure 25). As an instance of implementation, the session flow key is computed once for each flow and cached in a so-called transmission flow key cache (TFKC). If the Mapper (security policy module) has determined that the datagrams in a particular flow require confidentiality, then the datagrams are encrypted after the session flow key is obtained. The FBS header is generated and inserted into the datagram. The datagram and FBS header assembly is then forwarded to lower layers for transport.

(2) Receiver. When the FBS datagram arrives at the destination the FBS header is retrieved for processing. The timestamp is checked and if it does not validate because of a probable replay attack, the datagram is discarded. If validation is obtained, the SFL is recovered, concatenated with the derived master key, and the result fed into the same keyed one-way hash function used by the sender used to obtain the flow key. Much like the sender, once a flow key is recovered for a particular flow, it is cached in a receive flow key cache (RFKC). The cached flow key is then used to validate the MAC and decrypt all datagrams belonging to that particular flow.

## E. BENEFITS

The FBS allows duplex operation—participating users are able to receive and send at the same time. The caching of the cryptographic keys allows faster processing of datagrams at both ends. Rekeying is accomplished by changing the sfl. Key assignment can be done on a per flow basis or down to per datagram basis if the payload requires very strong protection. Since the master key is never transmitted and is never used to directly encrypt traffic, it is not susceptible to brute force attacks on captured traffic. Lastly, the compromise of one flow key does not place other flows at risk.

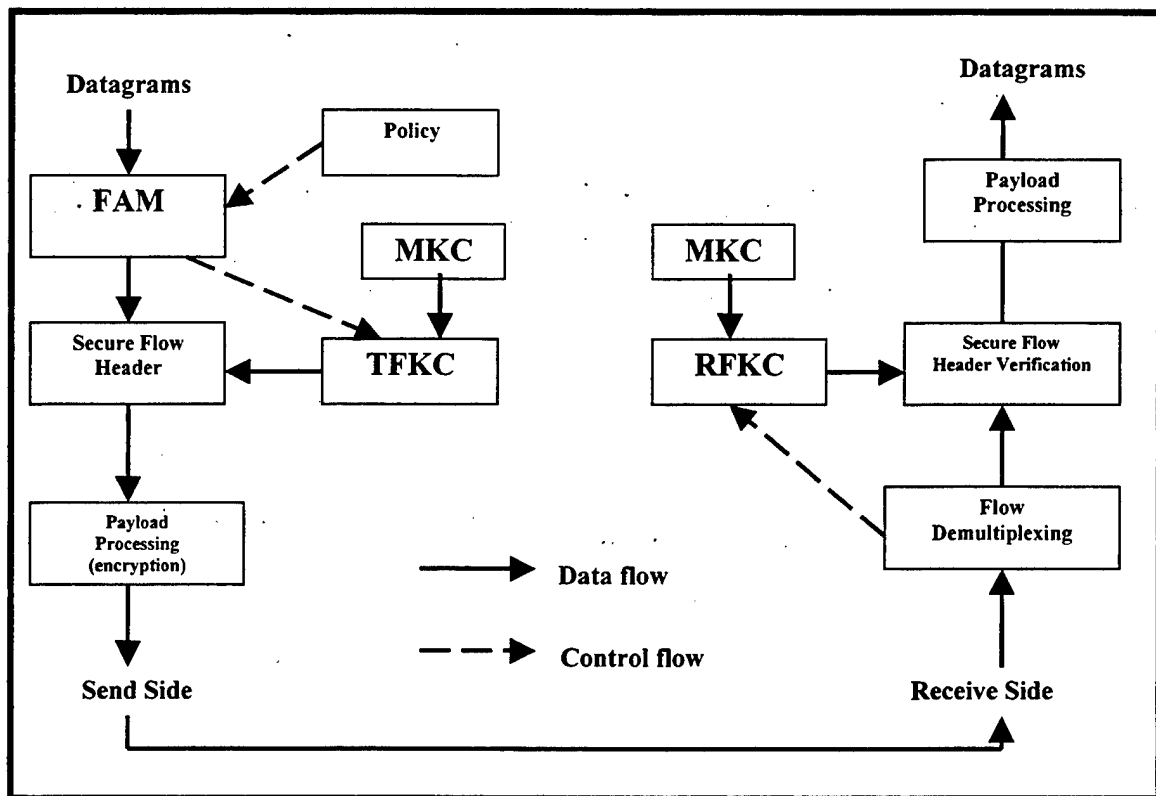


Figure 27. FBS protocol architecture and Operation  
From (Mitra and Woo 1997)

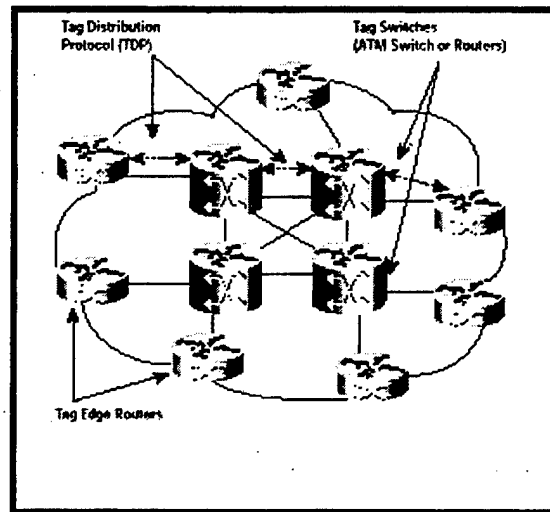
## **F. TAG SWITCHING**

Cisco Systems proposes an integrated solution of combining Link Layer Switching's superior throughput performance with the scalability of Network Layer Routing for enterprise networking. Tag switching is designed to couple the throughput performance of ATM switches with the network topologies of IP routers to reduce the overhead associated with forwarding IP packets. Central to Cisco's approach is the use of a tag attached to each multiprotocol data packet that serves as the routing information processed in Layer 2. Tags are short, fixed length labels which allows for simple and fast routing table lookups. Tag switching is primarily a software upgrade, which enhances its backward compatibility with current standards/protocol, and makes it economically attractive to network administrators.

### **1. Tag Edge Routers**

The Tag Switching network has three major elements: Tag edge routers (TER), tag switches (TS) and tag distribution protocol (TDP). Tag edge routers are the primary integrating elements between a homogenous ATM network and a heterogeneous IP network in tag switching. Located at the boundaries of IP-based networks, TERs are full-functioning Network Layer routers which use standard Internet routing protocols (e.g., EIGRP, BGP, OSPF) to determine routes through the IP network. (Cisco 1998). The resulting routing tables are used to assign and distribute tag information with TS using TDP. TERs take the TDP information to assemble a forwarding database, which utilizes tags. Upon arrival, the headers of incoming IP packets are examined by TERs for the destination address, and matched to a destination prefix entry in the tag-based routing

database. The proper tags are then applied to each packet (in the headers for IPv4 and in the flow label field for IPv6) and the packets are forwarded to the next routing destination based on the attached tag. This technique of mapping tags to packets provides the following flexibility:



**Figure 28.** Tag switching network  
From (Rechter, Davie et al. 1997)

- o Allows multiple sourced traffic destined to the same end host to share the same tag, thus economizing on the number of tags required and provide scalability.
- o Allows network managers to better manage loads between nodes, or respond favorably to unbalanced network topologies during node outage by tagging packets to flow along specified routes.

- o Allows finer granularity treatment such as that required for providing quality of service (QoS) (e.g., to a flow set up by the ReSource Reservation Protocol (RSVP)) when processing tagged packets.

Standard IP routers appear as switches to tag-equipped ATM switches when they are outfitted with tag switching software.

## **2. Tag Switches**

Tag switches form the core of a tag switching network. They provide the bridge to traffic tuning capabilities (Cisco 1998) for IP routers participating in the tag switching network. Tag switches implement Network Layer routing protocols along with TDP and ATM Forum signaling standards. The key difference between standard ATM and tag switching is the lack of connection set up procedure to allocate VCI's in switching (see section II.C.1). Therefore tag switching avoids the use of SVC associated with call setup, and sets free the ATM CPU processing capability in servicing longer-lived ATM virtual circuits (e.g., voice or video flows). Instead, tag switching uses standard IP routing protocol and TDP. The outcome is that tag switches are not burdened by the high call set up rates. A tag switch uses a tag of an incoming IP packet as an index in its Tag Information Base (TIB). The TIB contains entries that consist of an incoming tag, and one or more sub entries of the form [outgoing tag, outgoing interface, outgoing link level information]. If a match is made, an outgoing tag is attached and forwarded to the appropriate outgoing interface within the switch. When the packet reaches the destination TER, the tag is removed and the packet routed to the next hop. The tags are placed in the VCI field and ATM switching is accomplished according to VCI values.

Much like ATM, the routing decision in tag switching is based on exact match algorithm using short, fixed length fields.

### **3. Tag Distribution Protocol (TDP)**

Tag Distribution Protocol decouples the tag distribution from the data flows. It is the means by which routing information is exchanged among TER's and TS's. Routing databases are generated by TER's and TS's using standard IP routing protocols. Neighboring TER's and TS's exchange and distribute tag values to each other using TDP. Therefore routes (expressed in the TIB) are established before packet transmissions traverse the network. The network topology approach allows all categories of packet flows (e.g., long and/or short-lived) to participate in tag switching.

Degradation of performance and QoS occurs when packets are sent to a Network Layer function separate from the ATM cell path for routing resolution. Tag switching avoids this by switching all packets at the tag level. However, according to Cisco, unless special tag processing hardware is made available, tag switching may be relegated to data only while video and voice remains the performance realm of standard ATM networks (Cisco 1998).

### **G. SUMMARY**

AH and ESP, as primary security mechanisms of IPsec, do allow implementation of security services for authentication, confidentiality, and integrity. When combined with the multiple cryptographic key scheme, the per-packet protection approach is very effective in protecting information streams at the IP level. The multiple cryptographic key substructure adds to the complexity and level of difficulty for a network intruder to

attack the confidentiality, authentication and integrity of the IP packet stream. If a cryptographic key is compromised, the number of IP packets affected is no less than one and no more than the total number of packets to which the particular key was applied. The number of packets per cryptographic key is determined by an *a priori* agreement between participating hosts or during the call set up phase of connectivity. This key exchange philosophy adds another layer of difficulty for the network intruder as cryptographic keys may change at random or be synchronized according to a predetermined schedule for the duration of the connectivity. IPsec offers additional flexibility to the user by making the choice of cryptographic algorithms independent of the protocol itself.

The security services of IPsec are provided at the IP level. Since the complexity of IP (Layer 3) processing for routing flexibility significantly reduces throughput, the security advantages provided by IPsec is tempered by the disadvantages of IP processing on throughput. Not only are routing decisions conducted at OSI Layer 3 among routing nodes along the path of travel, security services processing is also executed at Layer 3.

ATM networks are able to process IP packets via the IP-over-ATM protocol. This provides IP packets with faster and highly reliable transport to the destination. However, the transition from IP format to ATM cell format and vice versa are latency points that exact a toll on the end-to-end throughput. With the exception of translating IP addresses to ATM addresses, ATM does not inspect every field of the IP header nor the IP payload. Therefore ATM does not care about and does not modify what is in the rest of the IP packet during transmission. This is both a strength for throughput and a glaring



weakness for security of Layer 2 switching. Thus when positioned between two IP nodes/network, ATM relies on the participating hosts and the IP layer to provide the needed security.

FBS has similarities to IPsec in its approach to providing information security services. It features origin authentication, integrity and confidentiality of message. Also, FBS provides flexibility in the application of cryptographic keys to individual IP packets, and in the selection of cryptographic algorithms when implementing the protocol. Where FBS differs from IPsec is in the area of demultiplexing media-specific, application-specific, or security level-specific information streams. While IPsec relegates the demultiplexing of received information to upper layer protocols (Transport layer, Session layer, or Application layer), FBS separates the IP packets according to "flows" by the use of a security flow label applied at the IP level. However, much like IP-based processing, FBS also suffers the disadvantages in throughput degradation in return for the security services provided.

All of the security and switching strategies presented in this chapter do not require modification of any existing Internet standard/protocol. Obviously modifications must be made on the host ends in order to comply with the correct processing of packets as specified by the strategy. None addresses the other elements of security services: Nonrepudiation, access control and availability. Essentially, the execution of these security services is entrusted to other components of the overall system. Based on the present degradation of throughput imposed by the current security strategy of IPsec and

FBS, the any additional security services processing at the IP level will further worsen the throughput situation.



### **III. FRAMEWORK FOR A LINK LAYER PACKET FILTERING (LLPF) SECURITY PROTOCOL**

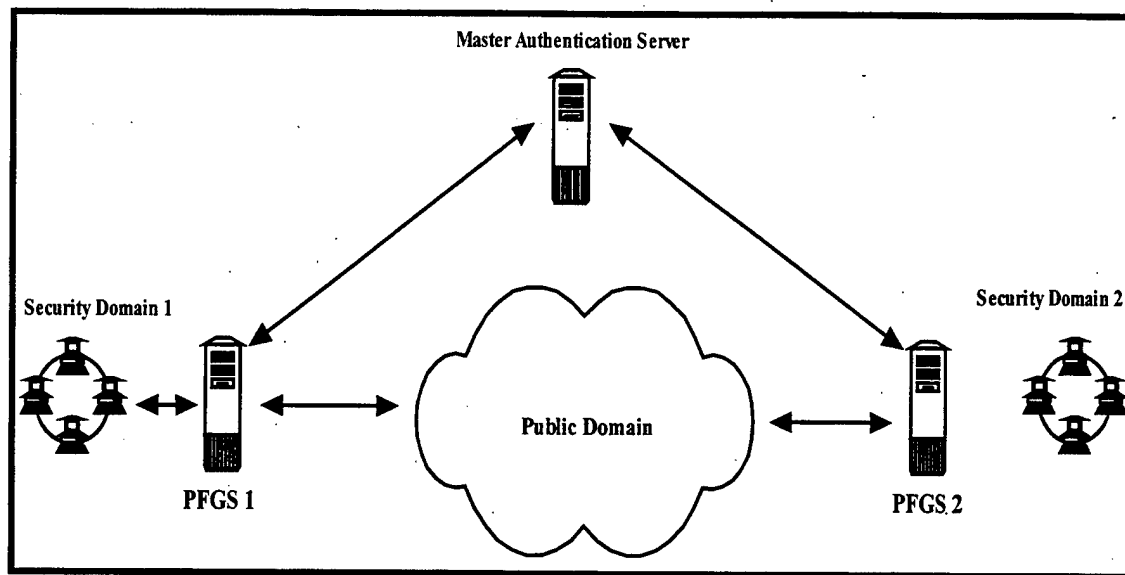
#### **A. INTRODUCTION**

This chapter describes the framework for a network protocol that provides security services while maintaining high throughput and routing flexibility. The framework is a packet filtering approach and has the following characteristics:

- o Security services (provided at the packet level) including data origin authentication and data integrity
- o Authentication Trailer (AT) appended to each packet, thus avoiding the overhead (latency) associated with the processing of complex IP headers
- o Processing of security data conducted at the Link Layer, thus maintaining high throughput
- o Use of short duration cryptographic keys, thus greatly minimizing successful brute force attacks on captured packets
- o Utilization of an automated key management scheme that supplies participating hosts with multiple session keys
- o Flexibility in the selection of cryptographic algorithms by the user
- o Use of the IP tunneling technique

- o Being compatible with current Internet standards/protocols and switching technology

The framework, which we call Link Layer Packet Filtering (LLPF), integrates the features that best meet the requirement of security with high throughput and routing flexibility, from Tag Switching, FBS, IPsec, and the IP protocol. However, LLPF boasts two innovations that set it apart from the security solutions described in Chapter II. These innovations are (1) the use of an authentication trailer and (2) multiple session keys of short duration. Figure 29 depicts our system model for design of a LLPF. The LLPF Security Protocol model uses a separate server for network user authentication and session key generation/distribution called Master Authentication Server (MAS). Packet filtering is conducted by a specially configured server called Packet Filtering Gateway Server (PFGS), which also serves as the gateway to the receiving/destination security domain.



**Figure 29.** System model for design of LLPF

## B. THE CONCEPT

### 1. Header VS Trailer

Current solutions such as IPsec rely on placing their security information and processing on the header end of IP packets. OSI Layer 3 (Network layer) processing requires that in order for a host (be it a router or a gateway ) to correctly process an IP packet, it must closely examine each of the individual fields in the IP header. It then becomes more convenient to have the security elements attached to the IP header (or placed before the IP header and after the IP payload) to share with the processing of the header fields. This tack is what makes security strategies such as IPsec and FSB compatible with current Internet standards/protocols. However, IPv4 and IPv6 formats both list a good number of immutable and mutable fields (see Figure 18, Chapter I and Appendix A and B) that must each be examined in order to fully and correctly process each IP packet. This close scrutiny of header fields for processing and routing decisions makes it very difficult to implement OSI Layer 3 switching/routing in hardware or firmware, and current implementation is restricted to software. This then is the crux of latency problem for IP level processing. If the additional fields (fixed and variable) of a security header are added on, we can safely conclude that the latency problem will be exacerbated. LLPF attaches the security processing information in a trailer - known as the *Authentication Trailer* (AT) - on the IP packet. With this approach, compatibility with current Internet standards/protocol is assured as the headers will remain as they are, ensuring routing through a heterogeneous IP network. Much like ATM cells, the AT have fixed fields, which reduces queuing delays (therefore reduced end-to-end network

latency) and makes switching more efficient (therefore high throughput). Once the filtering process has completed, the AT is removed and the packet switched to the next hop/processing node within the security domain.

## **2. In-Band VS Out-of-Band Call Setup and Connection Management**

Call setup and connection management under current Internet standards are conducted in the same virtual connection as the transmission of information. This is called in-band signaling. The deficiency of in-band signaling is that connection maintenance packets must compete for bandwidth along with information packets. For connection-less mode communication such as IP, this may be a drawback that must be accepted as cost of operation. However, in-band signaling does extract a security cost in that an intruder can obtain needed "hacking" information much easier by just searching for and monitoring one virtual connection for call setup, connection maintenance, along with plain data. ATM uses an out-of-band signaling method, which isolates call setup and connection maintenance from data transmission by use of dedicated virtual circuits. This approach ensures that priorities between service and maintenance bit streams and data bit streams do not compete with each other during switching. IP-based network uses the Transmission Control Protocol (TCP) (see Appendix D for additional details) for connection setup and management. TCP is a Transport/Session Layer-based, connection-oriented, end-to-end reliable protocol that provides reliable connection management services to IP-based network. By using a separate TCP session for call setup and cryptographic key management, we can simulate the "out-of-band" signaling infrastructure of ATM signaling. The separation of TCP connections for data and call-



setup-key-management improves reliability and assist in maintaining throughput by the use of dedicated “channels” for each function. In multi-LLPF sessions, security is enhanced by the use of out-of-band signaling as a level of difficulty is added in associating a service management TCP session with the right data TCP connection.

To reduce the overhead associated with call setup for authentication, the procedure is conducted once only on the following occasions:

- o A remote LLPF gateway is initiating a secure session with another compliant LLPF gateway for the very first time
- o The expiration of an authentication certificate (assuming the authentication protocol utilized issues certificates)

Likewise, key distribution occurs only on the following occasions:

- o A remote LLPF gateway has been authenticated for a secure session with another compliant LLPF gateway for the first time.
- o An active key table is about to or has cycled through completely.
- o The expiration of an authentication certificate, thus requiring another call setup authentication procedure to be conducted

Thus the call setup and key management procedures for our LLPF security protocol do not contribute to network congestion and overall end-to-end latency, but still provide the security services required of them.

### 3. Cryptographic Key Management

IPsec and FBS recommends a combination of asymmetric keying (public-key) and symmetric keying (shared secret key) as its instruments of cryptographic key distribution. Both rely on a session key for cryptographic computations in support of security functions. IPsec leaves it to the local security policy and user to determine the particular key exchange protocol to be implemented (e.g., Kerberos, Diffie-Hellman). FBS recommends the basic Diffie-Hellman key exchange model. LLPF abides by the same philosophy in that the selection of a particular key exchange model is the choice of users and their security policy.

The attractive element of the FBS keying procedure is that the session key (Master Key) is never transmitted, thereby enhancing security. LLPF uses a similar procedure with modifications. LLPF uses session keys for cryptographic computations but the keys themselves are generated and transmitted by a separate server called the *Master Authentication Server* (MAS). Instead of one key for the duration of the secure session, LLPF has the use of multiple keys per session, each key to be applied at a specified - but short - duration. An example is a table of 25 keys with each key used for a transmission duration of 15 seconds. This means that the key table takes 6.5 minutes to cycle through all the keys once and needs to be refreshed at the end of 6.5 minutes after the initial use of the key table. In the unlikely event that an intruder recovers a key from captured packets, only those packets that used the recovered key are compromised. Unless the recovered key cycles back for use again before the MAS refreshes the key table, intruders do not have access to the packets beyond the recovered key.

The MAS issues and refreshes the table of session keys on the following occasions:

- o A remote LLPF gateway has been authenticated for a secure session with another compliant LLPF gateway for the first time.
- o An active key table is about to or has cycled through completely.
- o The expiration of an authentication certificate, thus requiring another call setup authentication procedure to be conducted

As a result, the MAS must distribute a large number of keys per table in order to ensure sufficient number of keys are available for multiple connections and reconnections. The drawback is the processing overhead caused by the transmission and receipt of such a large key table. However, the advantage is that replacement for a new key table happens much less frequent, thus amortizing the initial penalty throughout the long life of a large key table. Appropriate computation procedures are established to prevent generation of identical key tables for two or more separate secure sessions. Just like FBS, the end hosts do not transmit the session keys. Instead, a key index, which points to a particular key in the key table, is sent as part of the AT. This, in effect, duplicates the key protection service provided by the FBS keying procedure.

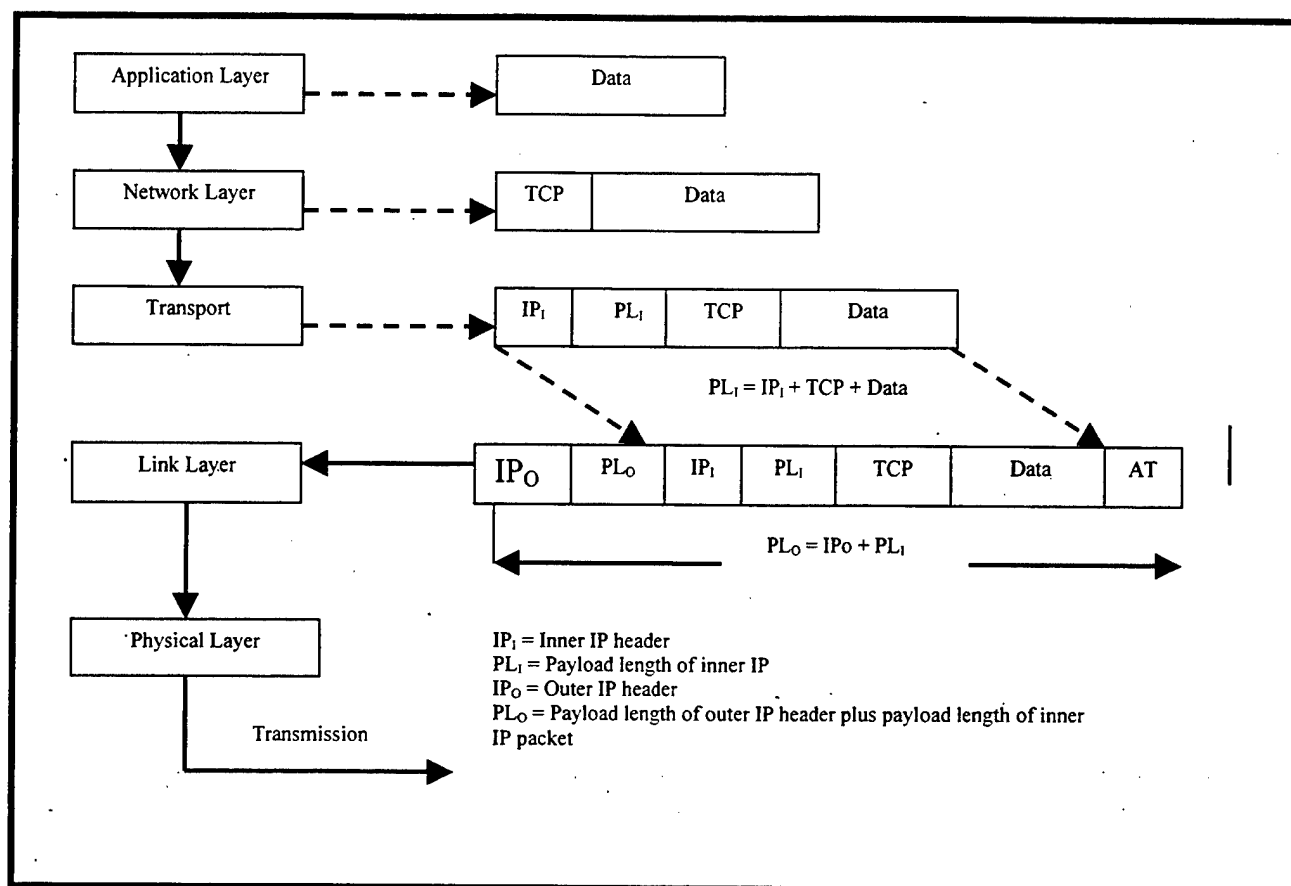
#### **4. Authentication Trailer and IP Routing**

A potential presented by attaching a trailer is that the packet length field value within the IP header no longer matches the actual packet length of the authenticated packet after the authentication trailer is stripped. This is so because, as previously

described, the authenticated packet – while in the LLPF gateway -- is never relayed to layer 3 where an updated packet length can be recomputed and attached. Therefore, according to the IP protocol, the destination host discards the authenticated packet once its layer 3 processing discovers the discrepancy. To alleviate this predicament, we employ the principle of tunneling by applying a modified version of IP Encapsulation (Perkins 1996).

According to RFC 2003, IP encapsulation is accomplished by appending another IP header before the original IP header and its payload. The appended outer IP header contains the source and a destination addresses. For this design, the outer source address is the same as the original source address in the inner IP header, while the outer IP destination address is the address for the destination packet filtering gateway server (PFGS). The original or inner IP destination address remains the address of the destination host within the destination security domain. The modification is to append the authentication trailer to the encapsulated IP. This addition is reflected only in the payload length field of the outer IP header. The payload length field value of the inner/original IP packet remains the same value as when the IP packet was created. Figure 30 illustrates the IP encapsulation .

When the new encapsulated packet construction is processed by the PFGS, the packet length field value of the inner IP header matches the actual bit length of the inner IP packet because the inner IP packet remains intact within the “tunnel” during its travel from the source to the destination. Therefore layer 3 of the destination’s protocol stack does not reject the packet.



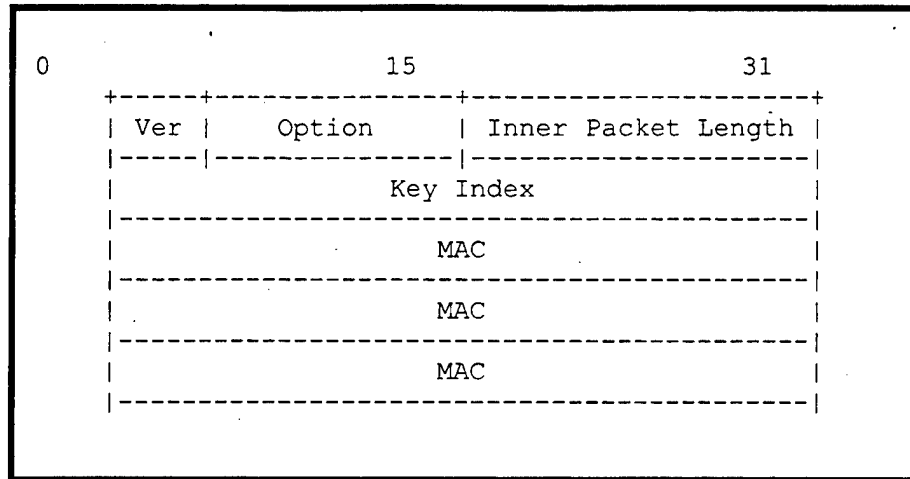
**Figure 30.** Encapsulation of IP packet with authentication trailer (AT)

## 5. Filtering Technique

The Ipsec AH protocol uses message digest (MD) of the IP packet to conduct data origin authentication, while FBS filters IP packets with message authentication code (MAC). FBS uses keyed one-way hash algorithms, which are considered to be far more secure than message digests. Therefore, for LLPF, packet filtering is accomplished by the use of MAC. The AT carries the MAC as a field element and uses it to authenticate all IP packets entering the security domain. The same is true for packets leaving the domain. Once the packet is authenticated, the PFGS strips the authentication trailer and

the outer IP header, and passes the remaining inner IP packet to the next hop for further routing/switching. The authenticated packet is never relayed to layer 3 for inspection of the IP header while in the PFGS, thus achieving layer 2 switching. The authentication trailer consists of the following (see Figure 31):

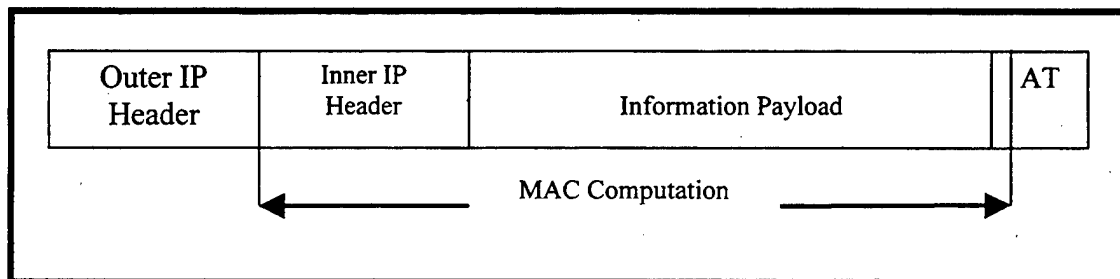
- o Version                                      3-bit fixed length field which identifies the version of the authentication trailer
- o Option                                        12-bit fixed length field reserved for future use
- o Inner Packet Length (IPL)            16-bit fixed length field which reflects the size of the inner packet (to include the authentication trailer)
- o Key Index                                    32-bit fixed length field which is the handle to the cryptographic key used to compute the MAC
- o Message Authentication Code (MAC)    Fixed length (128-bit) field that stores the MAC



**Figure 31.** Authentication Trailer format

The message authentication code is calculated over the entire inner IP packet and the first two words of the AT (see Figure 32). In this fashion, any modification to any portion of the inner packet will be detected and the packet discarded. Since the inner packet is not examined along the path of travel until arrival at its final destination, none of the fields in the header should change and therefore the attached MAC should remain valid upon arrival.

The length of the MAC is dependent on the type of protection required and the hashing algorithm choice.



**Figure 32.** Tunneled IP packet with AT

As previously stated, the key index is four bytes long and is transmitted unencrypted. Padding bits are used in the key index field (as well as in the IPL) in order to fix its length for all transmissions. The desire to ensure that both MAC and key index fields remain fixed is motivated by the high throughput possible in Layer 2 processing with fixed header and payload by ATM. Certainly the unencrypted view provides a potential attacker access to the information in the key index field. However, unless the attacker is able to decrypt and obtain the session key table, the key index information is useless to him/her. The attacker's most potent weapon against the use of trailers for filtering packets is to corrupt the trailer information (namely the key index) for many of the packets, thus forcing the PFGS to discard them. In this instance the attacker executes a denial-of-service attack. However, our security protocol is much more able to handle denial-of-service attacks because of its high throughput capability.

The IPL allows LLPF processing to immediately determine which section of the tunneled IP packet is the inner IP packet. Since the AT is a fixed length trailer, LLPF is able to extract it immediately from the tunneled IP packet by counting that segment of bits equal to the length of the AT from the opposite end of packer header location. Upon processing the AT/IPL, LLPF is then able to extract the complete segment of the inner IP packet by using the same methodology with which it processed the AT. The remaining segment after the extraction of the inner IP packet and the AT is the outer IP header, and it, along with the AT, is discarded after LLPF has completed the authentication process.

Connection maintenance packets of upper layer protocols such as ICMP and TCP are vulnerable to exploitation by intruders as well. ICMP packets can be used to map



network nodes within a security domain if the packets are not carefully screened. The same holds true for flow control packets from TCP. PFGS-filtering via AT can be extended in a straightforward fashion to handle upper layer connection maintenance packets.

## **6. Authentication and Filtering Servers**

The MAC is generated by the use of a keyed one-way hash function such as MD5. Cryptographic key distribution and management will occur separately but coordinated with data transfer and packet filtering process. That is, session set up and key distribution/management will be independent of information transfer, carried out in a separate and dedicated TCP session (akin to out-of-band signaling structure such as ATM). Cryptographic key distribution and management relies on a Master Authentication Server (MAS), and a Packet Filtering Gateway Server (PFGS), and occurs in two phases: user authentication and key table distribution.

### ***a. Master Authentication Server (MAS)***

The MAS is a stand-alone server that administers user authentication during set up. The MAS perform two functions: (1) authentication of end hosts, and (2) issuing shared session keys. The MAS manages a database containing all authorized users (remote/mobile), and maintains the set of cryptographic algorithms along with authentication and key exchange procedures as prescribed by the security policy. Authentication protocols such as Kerberos (Stallings 1998) and Netscape Communication's Secure Socket Layer (SSL) (Freier, Karlton et al. 1996) are candidates for the MAS' authentication and key exchange function.

The key table is generated by the MAS and distributed to authenticated PFGS's for use throughout the duration of the session. The key length, number of keys in the table, and the frequency of updates to the key table are balanced with the level of sensitivity of the data and desired level of security. The security policy in effect may not necessarily be stated at this level of specificity, but may provide guidelines to assist in the administrative configuration. Since the key table distribution requires security at least as rigorous as that needed for the most sensitive data, the choice of key exchange model should include a very secure distribution of a separate session key to encrypt/decrypt the key table during call set up and scheduled updates. In other words, a highly secure encryption algorithm should be utilized for the key table distribution.

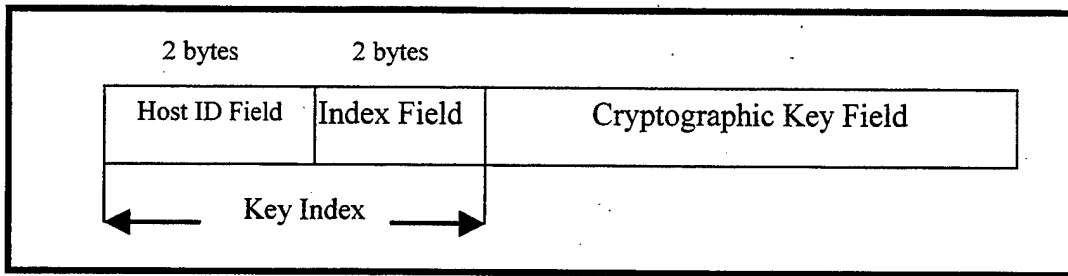
***b. Packet Filtering Gateway Server (PFGS)***

The PFGS utilizes the table of shared session keys to re-compute the MAC for each incoming IP packets and compare it to the attached MAC in the AT. All traffic routes through the PFGS and end host-to-end host secure sessions are conducted in TCP sessions separate from the key distribution/management. Those packets that do not have valid MACs are discarded. Data encryption and decryption are accomplished in the higher layer protocols.

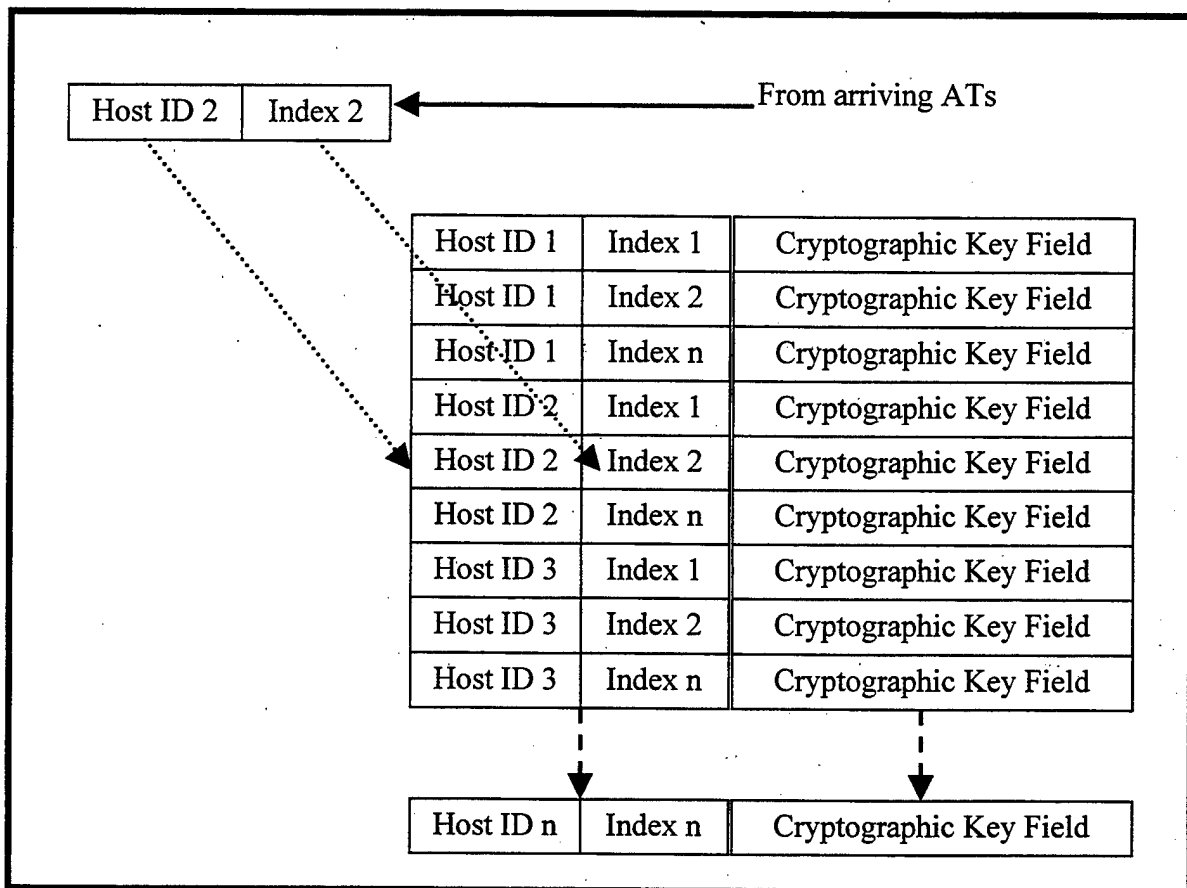
**7. Cryptographic Key table**

The key table uses a two-byte index subfield, which provides it with up to a maximum of 65K key selections. The user's security policy or the sensitivity of the data influences the length of the keys. The other two bytes in the index field represent the host identification to which the key index belongs. The host ID helps the processing

PFGS to separate the key table of two or more remote PFGS, thus assisting the processing PFGS to apply the matching key table to an active session or sessions. The host ID is generated by the receiving PFGS and relayed to the appropriate remote PFGS via the MAS. The two-byte length of the Host ID subfield provides the receiving PFGS with a selection of 65K Ids. The index field is appended to the key field as shown in Figure 33 and a reconstructed key table is shown in Figure 34.



**Figure 33.** A key with index



**Figure 34.** Sample key table

The “look-up” algorithm in Layer 2 processing would match the index field carried in the AT of the incoming packet to the four-byte index field column of the key table. The appended key of the matched index would then be used to generate a MAC for comparison with the attached MAC in the AT.

## **8. Packet Fragmentation**

The addition of the authentication trailer may appear to pose a problem for IP packet forwarding/routing. In practice, routers in route view the addition of the AT as transparent since any bit fields after the IP header appear to be just part of the total payload. Assuming that routers along the chosen path in the heterogeneous network comply with flow-based switching or Tag Switching, then the potential problem cited above becomes even more distant. If, however, there are a number of routers along the heterogeneous network path that do not exercise flow-based switching or Tag Switching, a potential defragmentation problem does exist. An IPv4 router may fragment a packet to a number of smaller packets to comply with the link maximum transmission unit (MTU) to the next node. This fragmentation does present a serious problem to the new packet filtering framework concept as the filtering gateway’s layer 2 will wait until all the smaller fragmented packets have arrived and are reassembled into the original transmitted packet before conducting the authentication process. In this situation, the fragmentation does add latency to the process. This latency can be avoided by specifying a packet size that is less than or equal to the path MTU prior to transmission. Nevertheless, this does not guarantee avoidance of defragmentation, as link MTU(s) may change due to dynamic routing changes caused by changing traffic conditions and router condition. Another

alternative for IPv4 is to set the Flags field to bit 1 (DF) for each packet, thus instructing routers along the network path to not fragment the packets. In contrast, IPv6 does not allow routers along the packet's delivery path to defragment the packet. Defragmentation can only occur at the source for IPv6. The IPv6 protocol requires that link MTUs be equal to or greater than 1280 bytes before defragmentation occurs.

### **C. OPERATION**

A secure session using LLPF and associated key distribution scheme would unfold as such:

Note: The assumption is that the PFGS 1 does not hold any key table and that this is the first time it has established a secure session with the PFGS 2.

#### **Phase 1:**

- a) PFGS 1, on behalf of a user within its security domain, establishes a TCP connection with the MAS and requests a secure IP session with PFGS 2.
- b) The MAS first verifies from its database that PFGS 1 is an authorized user. If not an authorized user, the authentication server immediately terminates the TCP connection. If an authorized user, authentication procedure is executed.
- c) Concurrently, the MAS verifies the PFGS 2 as an authorized user and authenticates PFGS 2.

- d) When PFGS 2 is authenticated, it sends an assigned Host ID to PFGS 1 via the MAS.
- e) Once both PFGS 1 & 2 are verified and authenticated, the MAS then proceeds with the key exchange model that would securely distribute the shared session key to decrypt the key table. Afterwards, the distribution of the encrypted key table to the participating PFGS's takes place.

Phase II:

- a) Upon receipt of the key table and the assigned Host ID, PFGS 1 starts the secure session by initiating a separate TCP connection to the destination host via PFGS 2.
- b) Using the key table, PFGS 1 computes the MAC of each packet, and attaches the AT containing the Host ID and key index in the Index Field to the packets.
- c) PFGS 1 transmit packets equipped with ATs to PFGS 2
- d) PFGS 2 authenticates packets from PFGS 1 by processing the ATs in Layer 2. If packets do not authenticate, they are discarded and an audit log entry is made.
- e) PFGS 2 forwards the authenticated packets to the next hop within its security domain for further routing toward the destination end host.
- f) The MAS updates or initiates another call setup procedure (authentication) if any of the conditions cited in III.B.2 & 3 occurs.

- g) Either PFGS, or the MAS terminates the session.

Establishment of follow-on sessions with PFGS 2 after termination of the first secure session occur as described in the following:

- o If the authentication certificates issued to PFGS 1 and PFGS 2 have not expired and the initial issue of key table has not cycled through completely, PFGS 1 may establish additional sessions with PFGS 2 directly without initiating another call setup and key table distribution procedure with the MAS.
- o If the authentication certificates issued to PFGS 1 and PFGS 2 have not expired but the previously issued key table has cycled through completely, PFGS 1 may request a new issue of the key table from the MAS without initiating a call setup procedure.
- o If either authentication certificate issued to PFGS 1 and PFGS 2 has expired, a call setup procedure with the MAS is initiated by the owner of the expiring certificate in order to obtain another authentication certificate and a new issue of the key table.

#### **D. KEY UTILIZATION AND MANAGEMENT**

The cryptographic key table provides necessary input values to the hashing function that must compute and produce the MAC. The use of multiple keys per session serves to increase proportionally the complexity of key discovery for potential attackers. This discovery complexity can be made more complex by increasing the size of the key



table, decreasing the period of use for each key, and refreshing/updating the key table more frequently. Increasing the bit-length of the key itself also serves to increase the complexity of discovery for an attacker. However, increasing the size of the key table, the key length and frequency of updates may add latency to key management as all of these transactions may add additional burden on the available bandwidth and processing capacity at both the ends of the connection. A large table may impose an additional time penalty to the look-up process in Layer 2 of the PFGS.

#### **E. POLICY**

In order to preserve the filtering utility of the AT in a secure session, the packet filtering function must occur at both ends of the connection. For a remote host belonging to a domain with a compliant PFGS, the AT filtering utility is preserved. For a remote host that does not have the use of a compliant PFGS and is using a dial-in connection via the Internet or a direct dial-in into the target security domain, the AT filtering utility is only provided at the target security domain using a compliant PFGS. In this instance, transmission of information may be unidirectional, from the remote host to the target PFGS. This inconvenience may be avoided if the proposed framework is implemented in hardware appropriate and compatible with portable computing devices of remote users.

#### **F. INTEROPERABILITY**

The proposed framework and its security mechanics are compatible with and do not require modification of current Internet standards/protocols. In conjunction with the use of AT packet filtering, the user may choose to implement IPsec (AH and/or ESP) and

other proposed IP-based security protocols for additional security services without fear of conflict. Choice of encryption algorithm, key management and authentication procedures depend upon configuration and data sensitivity. There are no modifications required for any hosts inside the security domain. The only modifications applied are to the PFGS' and the network stacks of portable/dial-in hosts.

## **G. SUMMARY**

Chapter II describes the specifics of the proposed framework for a packet filtering technique that uses authentication trailers (AT) instead of IP headers. The use of AT allows filtering to occur within Layer 2 thus permitting switching at Layer 2 as well. The fixed length feature of the AT fields makes hardware implementation feasible, thus achieving throughput normally associated with Layer 2 switching while providing security services of the quality usually associated with upper layers. The use of out-of-band signaling for call set up (user verification and authentication) and key management is not unique to the proposed framework and serves as an additional measure of security. The AT packet filtering framework is compatible to and does not require modification of current Internet standards/protocols.



## **IV. CONCLUSIONS AND RECOMMENDATIONS**

### **A. INTRODUCTION**

This chapter draws conclusions about Link Layer Packet Filtering. Section B presents our conclusion and Section C presents recommendations for additional and future work.

### **B. RESEARCH CONCLUSIONS**

#### **1. IP-Based Security Approach**

AH and ESP – as primary security mechanisms of IPsec – and FBS provides security services for authentication, confidentiality, and integrity. When combined with the multiple cryptographic key scheme, the per-packet protection approach is very effective in protecting information streams at the IP level. The multiple cryptographic key substructure adds to the complexity and level of difficulty for a network intruder to attack the confidentiality, authentication and integrity of the IP packet stream. If a cryptographic key is compromised, the number of IP packets affected is no less than one and no more than the total number of packets to which the particular key was applied. The IP-based security solutions offer additional flexibility to the user by making the choice of cryptographic algorithms independent of the protocol itself. In addition, current Internet standards/protocols are not modified in order to accommodate their security solution. However, IP-based security services are provided at the IP level. Since the complexity of IP (Layer 3) processing for routing flexibility significantly reduces throughput, the security advantages provided by IPsec is tempered by the disadvantages

of IP processing on throughput. Not only are routing decisions conducted at OSI Layer 3 among routing nodes along the path of travel, security services processing is also executed at Layer 3.

## **2. ATM and Tag Switching**

ATM networks are able to process IP packets via the IP-over-ATM protocol. This provides IP packets with a faster and highly reliable transport to the destination. However, the transition from IP format to ATM cell format and vice versa are latency points that exacts a toll on the end-to-end throughput. With the exception of translating IP addresses to ATM addresses, ATM does not inspect every field of the IP header nor the IP payload. Therefore ATM does not care about and does not modify what is in the rest of the IP packet during transmission. This is both a strength for throughput and a glaring weakness for security of Layer 2 switching. Thus when positioned between two IP nodes/network, ATM relies on the participating hosts and the IP layer to provide the needed security.

Tag Switching offers a solution to the latency problem associated with Layer 3 forwarding/routing of IP packets and its transition to and from ATM cells when traversing an ATM domain. It is an integrated solution of combining Link Layer Switching's superior throughput performance with the scalability of Network Layer Routing for enterprise networking. Tag switching is designed to couple the throughput performance of ATM switches with the network topologies of IP routers to reduce the overhead associated with forwarding IP packets. With regards to system administration, Tag Switching is a convenient and economical implementation because it is primarily a

software upgrade, which enhances its backward compatibility with current Internet standards/protocols. Much like ATM, Tag Switching does not offer security services for IP packets.

### **3. Link Layer Packet Filtering (LLPF) Security Protocol**

The use of a trailer vice a header to carry the information for authentication and integrity security services makes it a far more promising IP security solution due to the following consequences:

- o Avoidance of Layer 3 processing thus switching packets at Layer 2
- o Simplicity and fixed length nature allows it to be more conducive to hardware/firmware implementation
- o Transparent to upper layer protocol, therefore wide compatibility with current Internet standards/protocols and ATM specifications

The use of multiple keys of short duration enhances security far more than can IPsec or FBS. Not only can LLPF provide throughput at a level approaching ATM, it also has the routing flexibility that is characteristic of IP-based protocols. The LLPF security protocol can be applied in conjunction with other solutions for IP throughput and security. For example, Tag Switching can be combined with LLPF without any incompatibility problems. Tag Switching applies its advantages in switching packets through an IP network while LLPF provides authentication and integrity security services. For additional security services such as data encryption, LLPF can be combined with ESP with no fear of incompatibilities or network routing problems. By no means is

LLPF a complete package at this stage. There are additional analysis and evaluation to be conducted as stated in the recommendation section.

### **C. RECOMMENDATIONS FOR FUTURE WORK**

A good amount of time was spent on research, analysis of Internet standards/protocols, and hands on education of a CISCO ATM switch towards the formulation of the LLPF Security Protocol. As a result very little was accomplished towards addressing all operating issues that arose during the course of this research. These issues do impact on overall performance and may invariably affect the final appearance of the LLPF Security Protocol.

#### **1. Call Setup and Key Management**

Additional research and evaluation is needed to determine the appropriate key exchange model to employ for call setup and key management. The following questions apply:

- o Which key exchange model provides a secure call setup and key management services without incurring processing overhead for participating PFGS?
- o Which processing segment of a candidate key exchange model can be implemented in hardware/firmware in order to minimize computational and processing overhead for participating PFGS?

- o Which key exchange model can provide synchronization of key tables for participating PFGS? If none, how should the synchronization of key table for participating PFGS be constructed?

## **2. Key Length, Key Table and Frequency of Updates**

Additional research and evaluation is needed to determine the optimum mix of key bit length, size of key table, and the frequency of updates to the key table. The following questions apply:

- o What is the relationship among the three cryptographic key properties cited above with regards to throughput and security?
- o Is there an optimum value for each of the key properties with regards to maximum throughput - maximum security, maximum throughput-minimum security, or minimum throughput-maximum security?
- o Create a reference chart as to the right combination of the three key properties associated with a particular level of throughput (e.g., text, video, voice) and level of security (e.g., authentication only, authentication + encryption, etc.)

## **3. PFGS**

Additional research and evaluation is needed to determine the optimum and maximum number of sessions per PFGS and MAS. Other unresolved issues include:

- o Are there any other factors besides the number of sessions that may/do affect performance of the PFGS and MAS?



- o What are the advantages/disadvantages to throughput and security of separating the MAS function from the PFGS , or logically combining authentication and key management with the PFGS?

#### **D. SUMMARY**

This chapter presents conclusions on the performance advantages of the Link Layer Packet Filter Security Protocol in the areas of throughput, security, and routing flexibility. Recommendations for continued research and evaluation on performance issues not directly addressed by this research project are also presented.

## **APPENDIX A. ISO OSI REFERENCE MODEL**

This appendix provides the fundamentals in understanding the concept of network interoperability. The following are selected excerpts from Chapter 1 of Cisco Systems' Internetworking Technology Overview (Cisco 1997) addressing OSI reference model.

---

### **A. INTRODUCTION**

This appendix explains basic internetworking concepts. The information presented here helps readers who are new to internetworking comprehend the technical material that makes up the bulk of this thesis. Sections on the Open System Interconnection (OSI) reference model, important terms and concepts, and key organizations are included.

### **B. OSI REFERENCE MODEL**

#### **1. Introduction**

Moving information between computers of diverse design is a formidable task. In the early 1980s, the International Organization for Standardization (ISO) recognized the need for a network model that would help vendors create interoperable network implementations. The OSI reference model, released in 1984, addresses this need.

The OSI reference model quickly became the primary architectural model for intercomputer communications. Although other architectural models (mostly proprietary) have been created, most network vendors relate their network products to the OSI

reference model when they want to educate users about their products. Thus, the model is the best tool available to people hoping to learn about network technology.

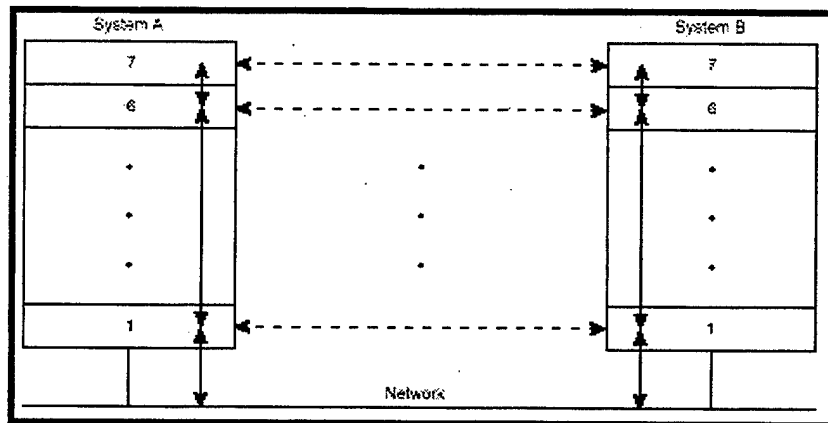
## **2. Hierarchical Communication**

The OSI reference model divides the problem of moving information between computers over a network medium into seven smaller and more manageable problems. Each of the seven smaller problems was chosen because it was reasonably self-contained and therefore more easily solved without excessive reliance on external information.

Each of the seven problem areas is solved by a *layer* of the model. Most network devices implement all seven layers. To streamline operations, however, some network implementations skip one or more layers. The lower two OSI layers are implemented with hardware and software; the upper five layers are generally implemented in software.

The OSI reference model describes how information makes its way from application programs (such as spreadsheets) through a network medium (such as wires) to another application program in another computer. As the information to be sent descends through the layers of a given system, it looks less and less like human language and more and more like the ones and zeros that a computer understands.

As an example of OSI-type communication, assume that System A in Figure 35 has information to send to System B. The application program in System A communicates with System A's Layer 7 (the top layer), which communicates with System A's Layer 6, which communicates with System A's Layer 5, and so on until

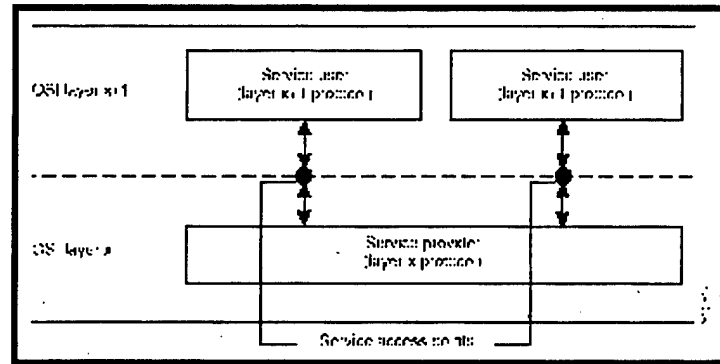


**Figure 35.** Communication between two computer systems

System A's Layer 1 is reached. Layer 1 is concerned with putting information on (and taking information off) the physical network medium. After the information has traversed the physical network medium and been absorbed into System B, it ascends through System B's layers in reverse order (first Layer 1, then Layer 2, and so on) until it finally reaches System B's application program. Although each of System A's layers communicates with its adjacent System A layers, its primary objective is to communicate with its peer layer in System B. That is, the primary objective of Layer 1 in System A is to communicate with Layer 1 in System B; Layer 2 in System A communicates with

Layer 2 in System B, and so on. This is necessary because each layer in a system has certain tasks it must perform. To perform these tasks, it must communicate with its peer layer in the other system. The OSI model's layering precludes direct communication between peer layers in different systems. Each layer in System A must therefore rely on services provided by adjacent System A layers to help achieve

communication with its System B peer. The relationship between adjacent layers in a single system is shown in Figure 36.



**Figure 36.** Relationship between adjacent layers in a single system

Assume Layer 4 in System A must communicate with Layer 4 in System B. To do this, Layer 4 in System A must use the services of Layer 3 in System A. Layer 4 is said to be the *service user*, while Layer 3 is the *service provider*. Layer 3 services are provided to Layer 4 at a *service access point* (SAP), which is simply a location at which Layer 4 can request Layer 3 services. As the figure shows, Layer 3 can provide its services to multiple Layer 4 entities.

### 3. Information Formats

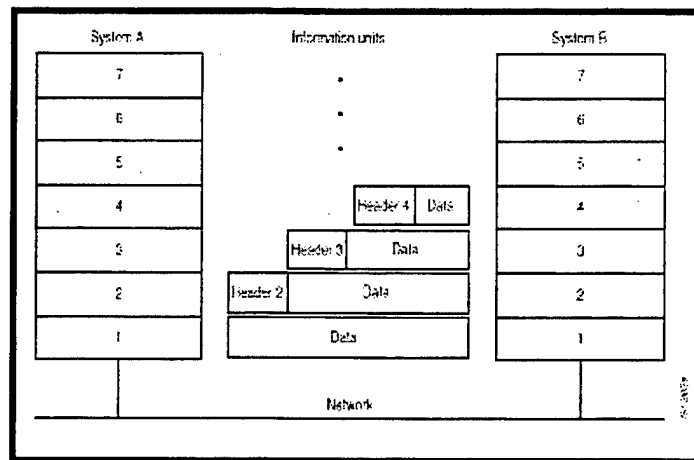
How does Layer 4 in System B know what Layer 4 in System A wants? Layer 4's specific requests are stored as *control information*, which is passed between peer layers in a block called a *header* that is prepended to the actual application information. For example, assume System A wishes to send the following text (called *data* or *information*) to System B:

The small grey cat ran up the wall to try to catch the red bird.

This text is passed from the application program in System A to System A's top layer.

System A's application layer must communicate certain information to System B's application layer, so it prepends that control information (in the form of a coded header) to the actual text to be moved. This information unit is passed to System A's Layer 6, which may prepend its own control information. The information unit grows in size as it descends through the layers until it reaches the network, where the original text and all associated control information travels to System B, where it is absorbed by System B's Layer 1. System B's Layer 1 strips the Layer 1 header, reads it, and then knows how to process the information unit. The slightly smaller information unit is passed to Layer 2, which strips the Layer 2 header, analyzes the header for actions Layer 2 must take, and so forth. When the information unit finally reaches the application program in System B, it simply contains the original text.

The concept of a header and data is relative, depending on the perspective of the layer currently analyzing the information unit. For example, to Layer 3, an information unit consists of a Layer 3 header and the data that follows. Layer 3's data, however, can potentially contain headers from Layers 4, 5, 6, and 7. Further, Layer 3's header is simply data to Layer 2. This concept is illustrated in Figure 37. Finally, not all layers need to append headers. Some layers simply perform a transformation on the actual data they receive to make the data more or less readable to their adjacent layers.



**Figure 37. Headers and data**

### **C. COMPATIBILITY ISSUES**

The OSI reference model is not a network implementation. Instead, it specifies the functions of each layer. In this way, it is like a blueprint for the building of a ship. After a ship blueprint is complete, the ship must still be built. Any number of shipbuilding companies can be contracted to do the actual work, just as any number of network vendors can build a protocol implementation from a protocol specification. And, unless the blueprint is extremely (impossibly) comprehensive, ships built by different shipbuilding companies using the same blueprint will differ from each other in at least minor ways. At the very least, for example, it is likely that the rivets will be in different places.

What accounts for the differences between implementations of the same ship blueprint (or protocol specification)? In part, the differences are due to the inability of any specification to consider every possible implementation detail. Also, different implementors will no doubt interpret the blueprint in slightly different ways. And, finally,

the inevitable implementation errors will cause different implementations to differ in execution. This explains why one company's implementation of protocol X does not always interoperate with another company's implementation of that protocol.

#### **D. OSI LAYERS**

Now that the basic features of the OSI layered approach have been described, each individual OSI layer and its functions can be discussed. Each layer has a predetermined set of functions it must perform for communication to occur.

##### **1. Application Layer**

The application layer is the OSI layer closest to the user. It differs from the other layers in that it does not provide services to any other OSI layer, but rather to application processes lying outside the scope of the OSI model. Examples of such application processes include spreadsheet programs, word-processing programs, banking terminal programs, and so on. The application layer identifies and establishes the availability of intended communication partners, synchronizes cooperating applications, and establishes agreement on procedures for error recovery and control of data integrity. Also, the application layer determines whether sufficient resources for the intended communication exist.

##### **2. Presentation Layer**

The presentation layer ensures that information sent by the application layer of one system will be readable by the application layer of another system. If necessary, the



presentation layer translates between multiple data representation formats by using a common data representation format.

The presentation layer concerns itself not only with the format and representation of actual user data, but also with data structures used by programs. Therefore, in addition to actual data format transformation (if necessary), the presentation layer negotiates data transfer syntax for the application layer.

### **3. Session Layer**

As its name implies, the session layer establishes, manages, and terminates sessions between applications. Sessions consist of dialogue between two or more presentation entities (recall that the session layer provides its services to the presentation layer). The session layer synchronizes dialogue between presentation layer entities and manages their data exchange. In addition to basic regulation of conversations (sessions), the session layer offers provisions for data expedition, class of service, and exception reporting of session-layer, presentation-layer, and application-layer problems.

### **4. Transport Layer**

The boundary between the session layer and the transport layer can be thought of as the boundary between application-layer protocols and lower-layer protocols. Whereas the application, presentation, and session layers are concerned with application issues, the lower four layers are concerned with data transport issues.

The transport layer attempts to provide a data transport service that shields the upper layers from transport implementation details. Specifically, issues such as how

reliable transport over an internetwork is accomplished are the concern of the transport layer. In providing reliable service, the transport layer provides mechanisms for the establishment, maintenance, and orderly termination of virtual circuits, transport fault detection and recovery, and information flow control (to prevent one system from overrunning another with data).

## **5. Network Layer**

The network layer is a complex layer that provides connectivity and path selection between two end systems that may be located on geographically diverse *subnetworks*. A subnetwork, in this instance, is essentially a single network cable (sometimes called a *segment*).

Because a substantial geographic distance and many subnetworks can separate two end systems desiring communication, the network layer is the domain of routing. Routing protocols select optimal paths through the series of interconnected subnetworks. Traditional network-layer protocols then move information along these paths.

## **6. Link Layer**

The link layer (formally referred to as the data link layer) provides reliable transit of data across a physical link. In so doing, the link layer is concerned with *physical* (as opposed to *network*, or *logical*) addressing, network topology, line discipline (how end systems will use the network link), error notification, ordered delivery of frames, and flow control.

## **7. Physical Layer**

The physical layer defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between end systems. Such characteristics as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, physical connectors, and other, similar, attributes are defined by physical layer specifications.

### **E. IMPORTANT TERMS AND CONCEPTS**

Internetworking, like other sciences, has a terminology and knowledge base all its own. Unfortunately, because the science of internetworking is so young, universal agreement on the meaning of networking concepts and terms has not yet occurred. Definitions of internetworking terms will become more rigidly defined and used as the internetworking industry matures.

#### **1. Addressing**

Locating computer systems on an internetwork is an essential component of any network system. There are various addressing schemes used for this purpose, depending on the protocol family being used. In other words, AppleTalk addressing is different from TCP/IP addressing, which in turn is different from OSI addressing, and so on.

Two important types of addresses are *link-layer* addresses and *network-layer* addresses. Link-layer addresses (also called *physical* or *hardware* addresses) are typically unique for each network connection. In fact, for most local-area networks (LANs), link-layer addresses are resident in the interface circuitry and are assigned by the organization

that defined the protocol standard represented by the interface. Because most computer systems have one physical network connection, they have only a single link-layer address. Routers and other systems connected to multiple physical networks can have multiple link-layer addresses. As their name implies, link-layer addresses exist at Layer 2 of the OSI reference model.

Network-layer addresses (also called *virtual* or *logical addresses*) exist at Layer 3 of the OSI reference model. Unlike link-layer addresses, which usually exist within a flat address space, network-layer addresses are usually hierarchical. In other words, they are like mail addresses, which describe a person's location by providing a country, a state, a zip code, a city, a street, an address on the street, and finally, a name. One good example of a flat address space is the U.S. social security numbering system, where each person has a single, unique social security number.

Hierarchical addresses make address sorting and recall easier by eliminating large blocks of logically similar addresses through a series of comparison operations. For example, we can eliminate all other countries if an address specifies the country Ireland. Easy sorting and recall is one reason that routers use network-layer addresses as the basis for routing.

Network-layer addresses differ depending on the protocol family being used, but they typically use similar logical divisions to find computer systems on an internetwork. Some of these logical divisions are based on physical network characteristics (such as the network segment a system is located on); others are based on groupings that have no physical basis (for example, the AppleTalk *zone*).

## 2. Frames, Packets, and Messages

Once addresses have located computer systems, information can be exchanged between two or more of these systems. Networking literature is inconsistent in naming the logically grouped units of information that move between computer systems. The terms *frame*, *packet*, *protocol data unit*, *PDU*, *segment*, *message*, and others have all been used, based on the whim of those who write protocol specifications.

In this publication, the term *frame* denotes an information unit whose source and destination is a link-layer entity. The term *packet* denotes an information unit whose source and destination is a network-layer entity. Finally, the term *message* denotes an information unit whose source and destination entity exists above the network layer. *Message* is also used to refer to particular lower-layer information units with a specific, well-defined purpose.

## **APPENDIX B. INTERNET PROTOCOL VERSION 4**

This appendix specifies the Department of Defense (DoD) Standard Internet Protocol, which is the basis for the Internet Standard 5. The following are excerpts from the Internet Request-For-Comments (RFC) document #791 (Information Sciences Institute 1981), DARPA Internet Protocol Program Protocol Specification.

---

### **A. INTRODUCTION**

#### **1. Motivation**

The Internet Protocol is designed for use in interconnected systems of packet-switched computer communication networks. Such a system has been called a "catenet". The internet protocol provides for transmitting blocks of data called datagrams from sources to destinations, where sources and destinations are hosts identified by fixed length addresses. The internet protocol also provides for fragmentation and reassembly of long datagrams, if necessary, for transmission through "small packet" networks.

#### **2. Scope**

The internet protocol is specifically limited in scope to provide the functions necessary to deliver a package of bits (an internet datagram) from a source to a destination over an interconnected system of networks. There are no mechanisms to augment end-to-end data reliability, flow control, sequencing, or other services commonly found in host-to-host protocols. The internet protocol can capitalize

on the services of its supporting networks to provide various types and qualities of service.

### **3. Interfaces**

This protocol is called on by host-to-host protocols in an internet environment. This protocol calls on local network protocols to carry the internet datagram to the next gateway or destination host. For example, a TCP module would call on the internet module to take a TCP segment (including the TCP header and user data) as the data portion of an internet datagram. The TCP module would provide the addresses and other parameters in the internet header to the internet module as arguments of the call. The internet module would then create an internet datagram and call on the local network interface to transmit the internet datagram.

### **4. Operation**

The internet protocol implements two basic functions: addressing and fragmentation. The internet modules use the addresses carried in the internet header to transmit internet datagrams toward their destinations. The selection of a path for transmission is called routing. The internet modules use fields in the internet header to fragment and reassemble internet datagrams when necessary for transmission through "small packet" networks. The model of operation is that an internet module resides in each host engaged in internet communication and in each gateway that interconnects networks. These modules share common rules for interpreting address fields and for fragmenting and assembling internet datagrams. In addition, these modules (especially in gateways) have procedures for making routing decisions and other functions.

The internet protocol treats each internet datagram as an independent entity unrelated to any other internet datagram. There are no connections or logical circuits (virtual or otherwise).

The internet protocol uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

The Type of Service is used to indicate the quality of the service desired. The type of service is an abstract or generalized set of parameters which characterize the service choices provided in the networks that make up the internet. This type of service indication is to be used by gateways to select the actual transmission parameters for a particular network, the network to be used for the next hop, or the next gateway when routing an internet datagram.

The Time to Live is an indication of an upper bound on the lifetime of an internet datagram. It is set by the sender of the datagram and reduced at the points along the route where it is processed. If the time to live reaches zero before the internet datagram reaches its destination, the internet datagram is destroyed. The time to live can be thought of as a self destruct time limit.

The Options provide for control functions needed or useful in some situations but unnecessary for the most common communications. The options include provisions for timestamps, security, and special routing.

The Header Checksum provides a verification that the information used in processing internet datagram has been transmitted correctly. The data may contain



errors. If the header checksum fails, the internet datagram is discarded at once by the entity which detects the error.

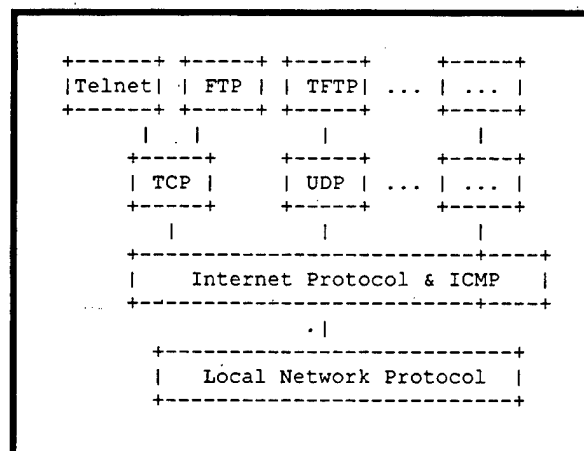
The internet protocol does not provide a reliable communication facility. There are no acknowledgments either end-to-end or hop-by-hop. There is no error control for data, only a header checksum. There are no retransmissions. There is no flow control.

Errors detected may be reported via the Internet Control Message Protocol (ICMP) which is implemented in the internet protocol module.

## B. OVERVIEW

### 1. Relation to Other Protocols

Figure 38 illustrates the place of the internet protocol in the protocol hierarchy:



**Figure 38.** Protocol relationships

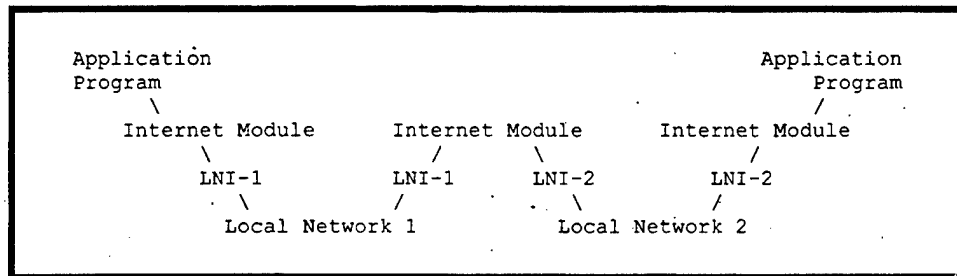
Internet protocol interfaces on one side to the higher level host-to-host protocols and on the other side to the local network protocol. In this context a "local network" may be a small network in a building or a large network such as the Internet.

## 2. Model of Operation

The model of operation for transmitting a datagram from one application program to another is illustrated (see Figure 39) by the following scenario:

- o We suppose that this transmission will involve one intermediate gateway.
- o The sending application program prepares its data and calls on its local internet module to send that data as a datagram and passes the destination address and other parameters as arguments of the call.
- o The internet module prepares a datagram header and attaches the data to it. The internet module determines a local network address for this internet address, in this case it is the address of a gateway. It sends this datagram and the local network address to the local network interface.
- o The local network interface creates a local network header, and attaches the datagram to it, then sends the result via the local network.
- o The datagram arrives at a gateway host wrapped in the local network header, the local network interface strips off this header, and turns the datagram over to the internet module. The internet module determines from the internet address that the datagram is to be forwarded to another host in a second network. The internet module determines a local net address for the destination host. It calls on the local network interface for that network to send the datagram.

- o This local network interface creates a local network header and attaches the datagram sending the result to the destination host.
- o At this destination host the datagram is stripped of the local net header by the local network interface and handed to the internet module.
- o The internet module determines that the datagram is for an application program in this host. It passes the data to the application program in response to a system call, passing the source address and other parameters as results of the call.



**Figure 39. Transmission path**

### **3. Function Description**

The function or purpose of Internet Protocol is to move datagrams through an interconnected set of networks. This is done by passing the datagrams from one internet module to another until the destination is reached. The internet modules reside in hosts and gateways in the internet system. The datagrams are routed from one internet module to another through individual networks based on the interpretation of an internet address. Thus, one important mechanism of the internet protocol is the internet address.

In the routing of messages from one internet module to another, datagrams may need to traverse a network whose maximum packet size is smaller than the size of the datagram. To overcome this difficulty, a fragmentation mechanism is provided in the internet protocol.

*a. Addressing*

A distinction is made between names, addresses, and routes. A name indicates what we seek. An address indicates where it is. A route indicates how to get there. The internet protocol deals primarily with addresses. It is the task of higher level (i.e., host-to-host or application) protocols to make the mapping from names to addresses. The internet module maps internet addresses to local net addresses. It is the task of lower level (i.e., local net or gateways) procedures to make the mapping from local net addresses to routes.

Addresses are fixed length of four octets (32 bits). An address begins with a network number, followed by local address (called the "rest" field). There are three formats or classes of internet addresses: in class a, the high order bit is zero, the next 7 bits are the network, and the last 24 bits are the local address; in class b, the high order two bits are one-zero, the next 14 bits are the network and the last 16 bits are the local address; in class c, the high order three bits are one-one-zero, the next 21 bits are the network and the last 8 bits are the local address.

Care must be taken in mapping internet addresses to local net addresses; a single physical host must be able to act as if it were several distinct hosts to the extent of

using several distinct internet addresses. Some hosts will also have several physical interfaces (multi-homing).

That is, provision must be made for a host to have several physical interfaces to the network with each having several logical internet addresses.

***b. Fragmentation***

Fragmentation of an internet datagram is necessary when it originates in a local net that allows a large packet size and must traverse a local net that limits packets to a smaller size to reach its destination.

An internet datagram can be marked "don't fragment." Any internet datagram so marked is not to be internet fragmented under any circumstances. If internet datagram marked don't fragment cannot be delivered to its destination without fragmenting it, it is to be discarded instead.

Fragmentation, transmission and reassembly across a local network which is invisible to the internet protocol module is called intranet fragmentation and may be used.

The internet fragmentation and reassembly procedure needs to be able to break a datagram into an almost arbitrary number of pieces that can be later reassembled.

The receiver of the fragments uses the identification field to ensure that fragments of different datagrams are not mixed. The fragment offset field tells the receiver the position of a fragment in the original datagram. The fragment offset and length determine the portion of the original datagram covered by this fragment. The

more-fragments flag indicates (by being reset) the last fragment. These fields provide sufficient information to reassemble datagrams.

The identification field is used to distinguish the fragments of one datagram from those of another. The originating protocol module of an internet datagram sets the identification field to a value that must be unique for that source-destination pair and protocol for the time the datagram will be active in the internet system. The originating protocol module of a complete datagram sets the more-fragments flag to zero and the fragment offset to zero.

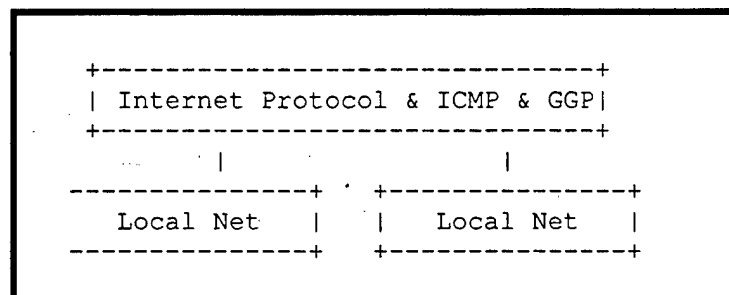
To fragment a long internet datagram, an internet protocol module (for example, in a gateway), creates two new internet datagrams and copies the contents of the internet header fields from the long datagram into both new internet headers. The data of the long datagram is divided into two portions on a 8 octet (64 bit) boundary (the second portion might not be an integral multiple of 8 octets, but the first must be). Call the number of 8 octet blocks in the first portion NFB (for Number of Fragment Blocks). The first portion of the data is placed in the first new internet datagram, and the total length field is set to the length of the first datagram. The more-fragments flag is set to one. The second portion of the data is placed in the second new internet datagram, and the total length field is set to the length of the second datagram. The more-fragments flag carries the same value as the long datagram. The fragment offset field of the second new internet datagram is set to the value of that field in the long datagram plus NFB.

This procedure can be generalized for an n-way split, rather than the two-way split described.

To assemble the fragments of an internet datagram, an internet protocol module (for example at a destination host) combines internet datagrams that all have the same value for the four fields: identification, source, destination, and protocol. The combination is done by placing the data portion of each fragment in the relative position indicated by the fragment offset in that fragment's internet header. The first fragment will have the fragment offset zero, and the last fragment will have the more-fragments flag reset to zero.

### c. *Gateways*

Gateways implement internet protocol to forward datagrams between networks. Gateways also implement the Gateway to Gateway Protocol (GGP) to coordinate routing and other internet control information. In a gateway the higher level protocols need not be implemented and the GGP functions are added to the IP module (see Figure 40).

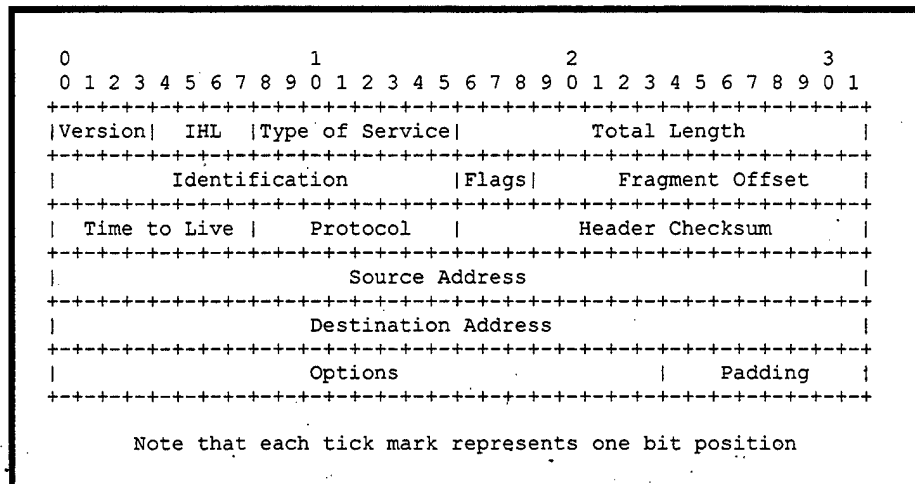


**Figure 40.** Gateway protocols

## C. SPECIFICATION

### 1. Internet Header Format

A summary of the contents of the internet header follows (see Figure 41):



**Figure 41.** Example of Internet datagram header

#### o Version: 4 bits

The Version field indicates the format of the internet header. This document describes version 4.

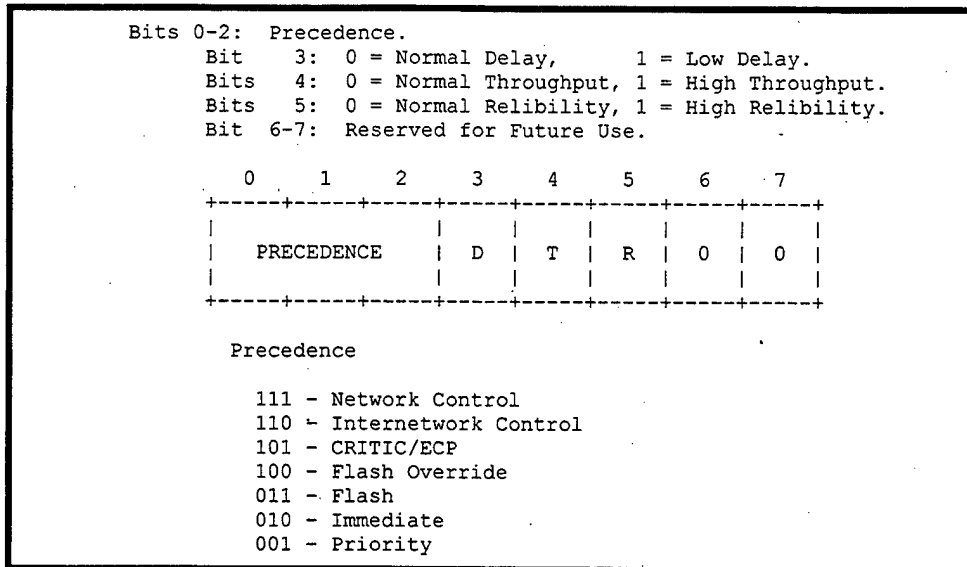
#### o IHL: 4 bits

Internet Header Length is the length of the internet header in 32 bit words, and thus points to the beginning of the data. Note that the minimum value for a correct header is 5.



- o Type of Service: 8 bits

The Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network. Several networks offer service precedence (see Figure 42), which somehow treats high precedence traffic as more important than other traffic (generally by accepting only traffic above a certain precedence at time of high load). The major choice is a three way tradeoff between low-delay, high-reliability, and high-throughput.



**Figure 42. Type-of-Service**

The use of the Delay, Throughput, and Reliability indications may increase the cost (in some sense) of the service. In many networks better performance for one of these parameters is coupled with worse performance on another. Except for very unusual cases at most two of these three indications should be set.

The type of service is used to specify the treatment of the datagram during its transmission through the internet system.

The Network Control precedence designation is intended to be used within a network only. The actual use and control of that designation is up to each network. The Internetwork Control designation is intended for use by gateway control originators only. If the actual use of these precedence designations is of concern to a particular network, it is the responsibility of that network to control the access to, and use of, those precedence designations.

- o Total Length: 16 bits

Total Length is the length of the datagram, measured in octets, including internet header and data. This field allows the length of a datagram to be up to 65,535 octets. Such long datagrams are impractical for most hosts and networks. All hosts must be prepared to accept datagrams of up to 576 octets (whether they arrive whole or in fragments). It is recommended that hosts only send datagrams larger than 576 octets if they have assurance that the destination is prepared to accept the larger datagrams.

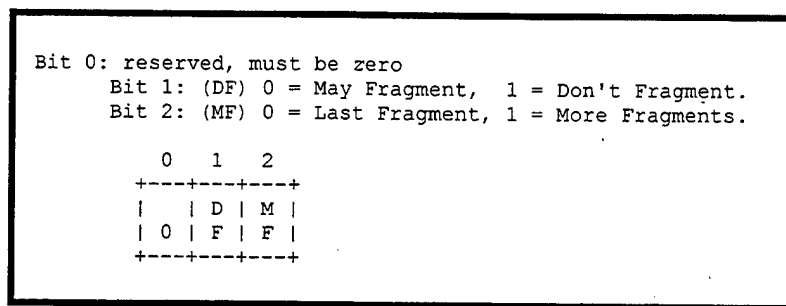
The number 576 is selected to allow a reasonable sized data block to be transmitted in addition to the required header information. For example, this size allows a data block of 512 octets plus 64 header octets to fit in a datagram. The maximal internet header is 60 octets, and a typical internet header is 20 octets, allowing a margin for headers of higher level protocols

- o Identification: 16 bits

An identifying value assigned by the sender to aid in assembling the fragments of a datagram.

- o Flags: 3 bits

Figure 43 is further illustration of control flags



**Figure 43.** Various control flags

- o Fragment Offset: 13 bits

This field indicates where in the datagram this fragment belongs. The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero.

- o Time to Live: 8 bits

This field indicates the maximum time the datagram is allowed to remain in the internet system. If this field contains the value zero, then the datagram must be destroyed. This field is modified in internet header processing. The time is measured in units of seconds, but since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.

- o Protocol: 8 bits

This field indicates the next level protocol used in the data portion of the internet datagram. The values for various protocols are specified in "Assigned Numbers" [Internet Standard 2

- o Header Checksum: 16 bits

A checksum on the header only. Since some header fields change (e.g., time to live), this is recomputed and verified at each point that the internet header is processed.

The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

This is a simple to compute checksum and experimental evidence indicates it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

- o Source Address: 32 bits

The source address.

- o Destination Address: 32 bits

The destination address. See section 3.2.

- o Options: variable

The options may appear or not in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation.

In some environments the security option may be required in all datagrams. The option field is variable in length. There may be zero or more options. There are two cases for the format of an option:

Case 1: A single octet of option-type.

Case 2: An option-type octet, an option-length octet, and the actual option-data octets.

The option-length octet counts the option-type octet and the option-length octet as well as the option-data octets. The option-type octet is viewed as having 3 fields:

1 bit copied flag,

2 bits option class,

5 bits option number.

The copied flag indicates that this option is copied into all fragments on fragmentation.

0 = not copied

1 = copied

The option classes are:

0 = control

1 = reserved for future use

2 = debugging and measurement

3 = reserved for future use

The following internet options are defined:

CLASS	NUMBER	LENGTH	DESCRIPTION
0	0	-	End of Option list. This option occupies only 1 octet; it has no length octet.
0	1	-	No Operation. This option occupies only 1 octet; it has no length octet.
0	2	11	Security. Used to carry Security, Compartmentation, User Group (TCC), and Handling Restriction Codes compatible with DOD requirements.
0	3	var.	Loose Source Routing. Used to route the internet datagram based on information supplied by the source.
0	9	var.	Strict Source Routing. Used to route the internet datagram based on information supplied by the source.
0	7	var.	Record Route. Used to trace the route an internet datagram takes.
0	8	4	Stream ID. Used to carry the stream identifier.
2	4	var.	Internet Timestamp.

#### D. IMPLEMENTATION

The implementation of a protocol must be robust. Each implementation must expect to interoperate with others created by different individuals. While the goal of this specification is to be explicit about the protocol there is the possibility of differing interpretations. In general, an implementation must be conservative in its sending behavior, and liberal in its receiving behavior. That is, it must be careful to send well-formed datagrams, but must accept any datagram that it can interpret (e.g., not object to technical errors where the meaning is still clear).

The basic internet service is datagram oriented and provides for the fragmentation of datagrams at gateways, with reassembly taking place at the destination internet protocol module in the destination host. Of course, fragmentation and reassembly of datagrams within a network or by private agreement between the gateways of a network is also allowed since this is transparent to the internet protocols and the higher-level protocols. This transparent type of fragmentation and reassembly is termed "network-dependent" (or intranet) fragmentation and is not discussed further here





## **APPENDIX C. INTERNET PROTOCOL VERSION 6**

This appendix describes the proposed new Internet Protocol new generation (version 6) specifications. The following are selected excerpts from the Internet-Draft, Internet Protocol Version 6 (IPv6) Specification (Deering and Hinden 1997).

---

### **A. INTRODUCTION**

IP version 6 (IPv6) is a new version of the Internet Protocol, designed as the successor to IP version 4 (IPv4) [RFC-791]. The changes from IPv4 to IPv6 fall primarily into the following categories:

- o Expanded Addressing Capabilities

IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable nodes, and simpler auto-configuration of addresses. The scalability of multicast routing is improved by adding a "scope" field to multicast addresses. And a new type of address called an "anycast address" is defined, used to send a packet to any one of a group of nodes.

- o Header Format Simplification

Some IPv4 header fields have been dropped or made optional, to reduce the common-case processing cost of packet handling and to limit the bandwidth cost of the IPv6 header.

- o Improved Support for Extensions and Options

Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

- o Flow Labeling Capability

A new capability is added to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling, such as non-default quality of service or "real-time" service.

- o Authentication and Privacy Capabilities

Extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv6.

## **B. TERMINOLOGY**

- o node - a device that implements IPv6.

- o router - a node that forwards IPv6 packets not explicitly addressed to itself. [See Note below].

- o host - any node that is not a router. [See Note below].

- o upper layer - a protocol layer immediately above IPv6. Examples are transport protocols such as TCP and UDP, control protocols such as ICMP,

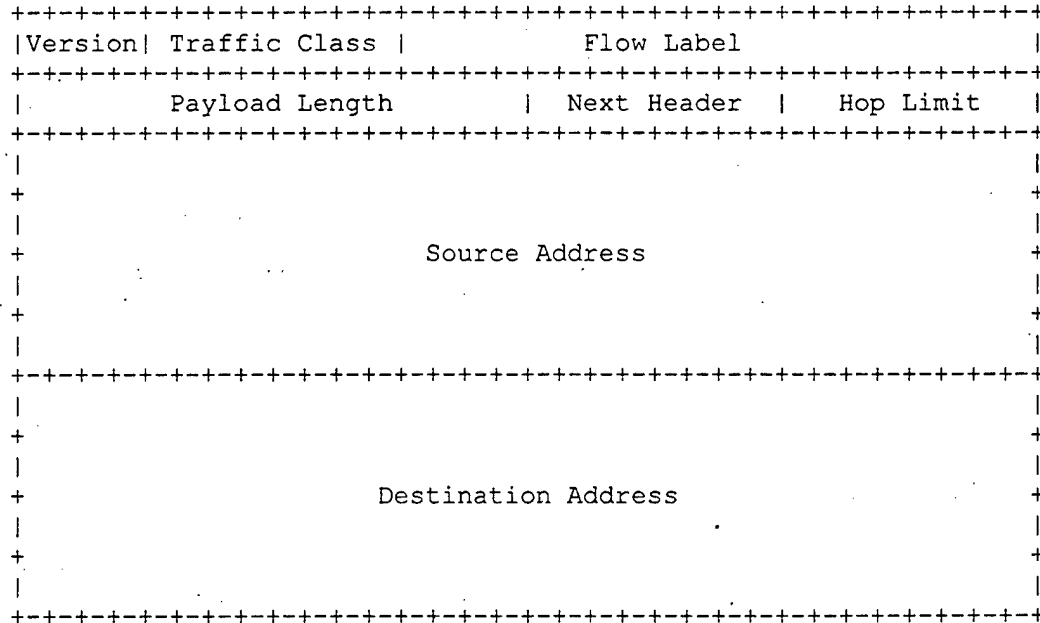
routing protocols such as OSPF, and internet or lower-layer protocols being "tunneled" over (i.e., encapsulated in) IPv6 such as IPX, AppleTalk, or IPv6 itself.

- o link - a communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernets (simple or bridged); PPP links; X.25, Frame Relay, or ATM networks; and internet (or higher) layer "tunnels", such as tunnels over IPv4 or IPv6 itself.
- o neighbors - nodes attached to the same link.
- o interface - a node's attachment to a link.
- o address - an IPv6-layer identifier for an interface or a set of interfaces.
- o packet - an IPv6 header plus payload.
- o link MTU - the maximum transmission unit, i.e., maximum packet size in octets, that can be conveyed over a link.
- o path MTU - the minimum link MTU of all the links in a path between a source node and a destination node.

Note: it is possible, though unusual, for a device with multiple interfaces to

be configured to forward non-self-destined packets arriving from some set (fewer than all) of its interfaces, and to discard non-self-destined packets arriving from its other interfaces. Such a device must obey the protocol requirements for routers when receiving packets from, and interacting with neighbors over, the former (forwarding) interfaces. It must obey the protocol requirements for hosts when receiving packets from, and interacting with neighbors over, the latter (non-forwarding) interfaces.

### C. IPV6 HEADER FORMAT



- o Version                      4-bit Internet Protocol version number = 6.

- o Traffic Class      8-bit traffic class field. See section 7.
- o Flow Label        20-bit flow label. See section 6.
- o Payload Length    16-bit unsigned integer. Length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets.

(Note that any extension headers present are considered part of the payload, i.e., included in the length count.)

- o Next Header        8-bit selector. Identifies the type of header immediately following the IPv6 header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
- o Hop Limit          8-bit unsigned integer. Decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.
- o Source Address     128-bit address of the originator of the packet.
- o Destination Address   128-bit address of the intended recipient of the packet (possibly not the ultimate recipient, if a Routing header is present).

## D. IPV6 EXTENSION HEADERS

In IPv6, optional internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper-layer header in a packet. There are a small number of such extension headers, each identified by a distinct Next Header value. As illustrated in these examples, an IPv6 packet may carry zero, one, or more extension headers, each identified by the Next Header field of the preceding header:

IPv6 header	TCP header + data
Next Header =	
TCP	

IPv6 header	Routing header	TCP header + data
Next Header =	Next Header =	
Routing	TCP	

IPv6 header	Routing header	Fragment header	fragment of TCP header + data
Next Header =	Next Header =	Next Header =	
Routing	Fragment	TCP	

With one exception, extension headers are not examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header.

There, normal demultiplexing on the Next Header field of the IPv6 header invokes the module to process the first extension header, or the upper-layer header if no extension

header is present. The contents and semantics of each extension header determine whether or not to proceed to the next header. Therefore, extension headers must be processed strictly in the order they appear in the packet; a receiver must not, for example, scan through a packet looking for a particular kind of extension header and process that header prior to processing all preceding ones.

The exception referred to in the preceding paragraph is the Hop-by-Hop Options header, which carries information that must be examined and processed by every node along a packet's delivery path, including the source and destination nodes. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.

If, as a result of processing a header, a node is required to proceed to the next header but the Next Header value in the current header is unrecognized by the node, it should discard the packet and send an ICMP Parameter Problem message to the source of the packet, with an ICMP Code value of 1 ("unrecognized Next Header type encountered") and the ICMP Pointer field containing the offset of the unrecognized value within the original packet. The same action should be taken if a node encounters a Next Header value of zero in any header other than an IPv6 header.

Each extension header is an integer multiple of 8 octets long, in order to retain 8-octet alignment for subsequent headers. Multi-octet fields within each extension header are aligned on their natural boundaries, i.e., fields of width  $n$  octets are placed at an integer multiple of  $n$  octets from the start of the header, for  $n = 1, 2, 4$ , or  $8$ .



A full implementation of IPv6 includes implementation of the following extension headers:

- o Hop-by-Hop Options
- o Routing (Type 0)
- o Fragment
- o Destination Options
- o Authentication
- o Encapsulating Security Payload

The first four are specified in this document; the last two are specified in [RFC-1826] and [RFC-1827], respectively.

#### **E. EXTENSION HEADER ORDER**

When more than one extension header is used in the same packet, it is recommended that those headers appear in the following order:

IPv6 header

Hop-by-Hop Options header

Destination Options header (note 1)

Routing header

Fragment header

Authentication header (note 2)

Encapsulating Security Payload header (note 2)

Destination Options header (note 3)

upper-layer header

note 1: for options to be processed by the first destination that appears in

the IPv6 Destination Address field plus subsequent destinations listed in the

Routing header.

note 2: additional recommendations regarding the relative order of the

Authentication and Encapsulating Security Payload headers are given in [RFC 1827].

note 3: for options to be processed only by the final destination of the packet.

Each extension header should occur at most once, except for the Destination Options header which should occur at most twice (once before a Routing header and once before the upper-layer header).

If the upper-layer header is another IPv6 header (in the case of IPv6 being tunneled over or encapsulated in IPv6), it may be followed by its own extension headers, which are separately subject to the same ordering recommendations. If and when other

extension headers are defined, their ordering constraints relative to the above listed headers must be specified.

IPv6 nodes must accept and attempt to process extension headers in any order and occurring any number of times in the same packet, except for the Hop-by-Hop Options header which is restricted to appear immediately after an IPv6 header only. Nonetheless, it is strongly advised that sources of IPv6 packets adhere to the above recommended order until and unless subsequent specifications revise that recommendation.

## F. OPTIONS

Two of the currently-defined extension headers -- the Hop-by-Hop Options header and the Destination Options header -- carry a variable number of type-length-value (TLV) encoded "options", of the following format:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| Option Type | Opt Data Len | Option Data |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

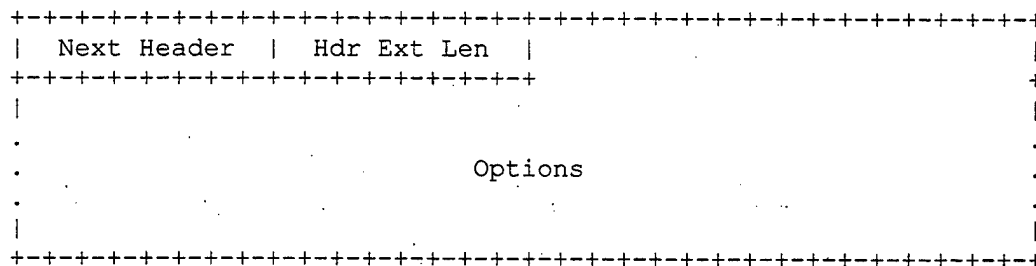
- o Option Type      8-bit identifier of the type of option.
- o Opt Data Len     8-bit unsigned integer. Length of the Option Data field of this option, in octets.
- o Option Data      Variable-length field. Option-Type-specific data.

The sequence of options within a header must be processed strictly in the order they appear in the header; a receiver must not, for example, scan through the header

looking for a particular kind of option and process that option prior to processing all preceding ones.

## G. HOP-BY-HOP OPTIONS HEADER

The Hop-by-Hop Options header is used to carry optional information that must be examined by every node along a packet's delivery path. The Hop-by-Hop Options header is identified by a Next Header value of 0 in the IPv6 header, and has the following format:



- o Next Header      8-bit selector. Identifies the type of header immediately following the Hop-by-Hop Options header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
- o Hdr Ext Len      8-bit unsigned integer. Length of the Hop-by-Hop Options header in 8-octet units, not including the first 8 octets.
- o Options          Variable-length field, of length such that the complete Hop-by-Hop Options header is an integer multiple of 8 octets long.

## H. ROUTING HEADER

The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination. This function is very similar to IPv4's Loose Source and Record Route option. The Routing header is identified by a Next Header value of 43 in the immediately preceding header, and has the following format:

```
+-----+
| Next Header | Hdr Ext Len | Routing Type | Segments Left |
+-----+
|
|                                     |
|                                     |
|                                     |
|                                     |
|                                     |
+-----+
```

- o Next Header      8-bit selector. Identifies the type of header immediately following the Routing header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq].
- o Hdr Ext Len      8-bit unsigned integer. Length of the Routing header in 8-octet units, not including the first 8 octets.
- o Routing Type      8-bit identifier of a particular Routing header variant.
- o Segments Left    8-bit unsigned integer. Number of route segments remaining, i.e., number of explicitly listed intermediate nodes still to be visited before reaching the final destination.

- o type-specific data Variable-length field, of format determined by the Routing Type, and of length such that the complete Routing header is an integer multiple of 8 octets long.

If, while processing a received packet, a node encounters a Routing header with an unrecognized Routing Type value, the required behavior of the node depends on the value of the Segments Left field, as follows:

- o If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet, whose type is identified by the Next Header field in the Routing header.
- o If Segments Left is non-zero, the node must discard the packet and send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

If, after processing a Routing header of a received packet, an intermediate node determines that the packet is to be forwarded onto a link whose link MTU is less than the size of the packet, the node must discard the packet and send an ICMP Packet Too Big message to the packet's Source Address.

## **I. FRAGMENT HEADER**

The Fragment header is used by an IPv6 source to send a packet larger than would fit in the path MTU to its destination. (Note: unlike IPv4, fragmentation in IPv6 is

performed only by source nodes, not by routers along a packet's delivery path -- see section 5.) The Fragment header is identified by a Next Header value of 44 in the immediately preceding header, and has the following format:

```

+-----+
| Next Header | Reserved | Fragment Offset | Res | M |
+-----+
| Identification |
+-----+

```

- o Next Header      8-bit selector. Identifies the initial header type of the Fragmentable Part of the original packet (defined below). Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
- o Reserved          8-bit reserved field. Initialized to zero for transmission; ignored on reception.
- o Fragment Offset   13-bit unsigned integer. The offset, in 8-octet units, of the data following this header, relative to the start of the Fragmentable Part of the original packet.
- o Res                2-bit reserved field. Initialized to zero for transmission; ignored on reception.
- o M flag             1 = more fragments; 0 = last fragment.
- o Identification    32 bits.

In order to send a packet that is too large to fit in the MTU of the path to its destination, a source node may divide the packet into fragments and send each fragment as a separate packet, to be reassembled at the receiver.

For every packet that is to be fragmented, the source node generates an Identification value. The Identification must be different than that of any other fragmented packet sent recently\* with the same Source Address and Destination Address. If a Routing header is present, the Destination Address of concern is that of the final destination.

\* "recently" means within the maximum likely lifetime of a packet, including transit time from source to destination and time spent awaiting reassembly with other fragments of the same packet. However, it is not required that a source node know the maximum packet lifetime. Rather, it is assumed that the requirement can be met by maintaining the Identification value as a simple, 32-bit, "wrap-around" counter, incremented each time a packet must be fragmented. It is an implementation choice whether to maintain a single counter for the node or multiple counters, e.g., one for each of the node's possible source addresses, or one for each active (source address, destination address) combination.

The following rules govern reassembly:

- o An original packet is reassembled only from fragment packets that have the same Source Address, Destination Address, and Fragment Identification.



- o The Unfragmentable Part of the reassembled packet consists of all headers up to, but not including, the Fragment header of the first fragment packet (that is, the packet whose Fragment Offset is zero), with the following two changes:

The Next Header field of the last header of the Unfragmentable Part is obtained from the Next Header field of the first fragment's Fragment header.

- o The Payload Length of the reassembled packet is computed from the length of the Unfragmentable Part and the length and offset of the last fragment. For example, a formula for computing the Payload Length of the reassembled original packet is:

$PL_{orig} = PL_{first} - FL_{first} - 8 + (8 * FO_{last}) + FL_{last}$  where

$PL_{orig}$  = Payload Length field of reassembled packet.

$PL_{first}$  = Payload Length field of first fragment packet.

$FL_{first}$  = length of fragment following Fragment header of first fragment packet.

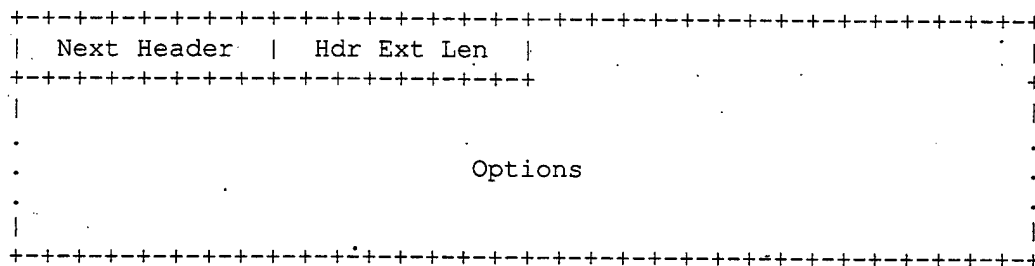
$FO_{last}$  = Fragment Offset field of Fragment header of last fragment packet.

$FL_{last}$  = length of fragment following Fragment header of last fragment packet.

The Fragmentable Part of the reassembled packet is constructed from the fragments following the Fragment headers in each of the fragment packets. The length of each fragment is computed by subtracting from the packet's Payload Length the length of the headers between the IPv6 header and fragment itself; its relative position in Fragmentable Part is computed from its Fragment Offset value.

## J. DESTINATION OPTIONS HEADER

The Destination Options header is used to carry optional information that need be examined only by a packet's destination node(s). The Destination Options header is identified by a Next Header value of 60 in the immediately preceding header, and has the following format:



- o **Next Header**      8-bit selector. Identifies the type of header immediately following the Destination Options header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
- o **Hdr Ext Len**      8-bit unsigned integer. Length of the Destination Options header in 8-octet units, not including the first 8 octets.

- o Options                      Variable-length field, of length such that the complete Destination Options header is an integer multiple of 8 octets long.

Note that there are two possible ways to encode optional destination information in an IPv6 packet: either as an option in the Destination Options header, or as a separate extension header. The Fragment header and the Authentication header are examples of the latter approach. Which approach can be used depends on what action is desired of a destination node that does not understand the optional information:

- o If the desired action is for the destination node to discard the packet and, only if the packet's Destination Address is not a multicast address, send an ICMP Unrecognized Type message to the packet's Source Address, then the information may be encoded either as a separate header or as an option in the Destination Options header whose Option Type has the value 11 in its highest-order two bits. The choice may depend on such factors as which takes fewer octets, or which yields better alignment or more efficient parsing.
- o If any other action is desired, the information must be encoded as an option in the Destination Options header whose Option Type has the value 00, 01, or 10 in its highest-order two bits, specifying the desired action (see section 4.2).

## **K. PACKET SIZE ISSUES**

IPv6 requires that every link in the internet have an MTU of 1280 octets or greater. On any link that cannot convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6. Links that have a configurable MTU (for example, PPP links [RFC-1661]) must be configured to have an MTU of at least 1280 octets; it is recommended that they be configured with an MTU of 1500 octets or greater, to accommodate possible encapsulations (i.e., tunneling) without incurring IPv6-layer fragmentation.

From each link to which a node is directly attached, the node must be able to accept packets as large as that link's MTU. It is strongly recommended that IPv6 nodes implement Path MTU Discovery [RFC-1981], in order to discover and take advantage of path MTUs greater than 1280 octets. However, a minimal IPv6 implementation (e.g., in a boot ROM) may simply restrict itself to sending packets no larger than 1280 octets, and omit implementation of Path MTU Discovery.

In order to send a packet larger than a path's MTU, a node may use the IPv6 Fragment header to fragment the packet at the source and have it reassembled at the destination(s). However, the use of such fragmentation is discouraged in any application that is able to adjust its packets to fit the measured path MTU (i.e., down to 1280 octets).

A node must be able to accept a fragmented packet that, after reassembly, is as large as 1500 octets. A node is permitted to accept fragmented packets that reassemble to more than 1500 octets. An upper-layer protocol or application that depends on IPv6

fragmentation to send packets larger than the MTU of a path should not send packets larger than 1500 octets unless it has assurance that the destination is capable of reassembling packets of that larger size.

In response to an IPv6 packet that is sent to an IPv4 destination (i.e., a packet that undergoes translation from IPv6 to IPv4), the originating IPv6 node may receive an ICMP Packet Too Big message reporting a Next-Hop MTU less than 1280. In that case, the IPv6 node is not required to reduce the size of subsequent packets to less than 1280, but must include a Fragment header in those packets so that the IPv6-to-IPv4 translating router can obtain a suitable Identification value to use in resulting IPv4 fragments. Note that this means the payload may have to be reduced to 1232 octets (1280 minus 40 for the IPv6 header and 8 for the Fragment header), and smaller still if additional extension headers are used.

## **L. FLOW LABELS**

The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. This aspect of IPv6 is, at the time of writing, still experimental and subject to change as the requirements for flow support in the Internet become clearer. Hosts or routers that do not support the functions of the Flow Label field are required to set the field to zero when originating a packet, pass the field on unchanged when forwarding a packet, and ignore the field when receiving a packet. The Appendix describes the current intended semantics and usage of the Flow Label field.

## **M. TRAFFIC CLASSES**

The 8-bit Traffic Class field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets. At the point in time at which this specification is being written, there are a number of experiments underway in the use of the IPv4 Type of Service and/or Precedence bits to provide various forms of "differentiated service" for IP packets, other than through the use of explicit flow set-up. The Traffic Class field in the IPv6 header is intended to allow similar functionality to be supported in IPv6. It is hoped that those experiments will eventually lead to agreement on what sorts of traffic classifications are most useful for IP packets. Detailed definitions of the syntax and semantics of all or some of the IPv6 Traffic Class bits, whether experimental or intended for eventual standardization, are to be provided in separate documents.

The following general requirements apply to the Traffic Class field:

- o The service interface to the IPv6 service within a node must provide a means for an upper-layer protocol to supply the value of the Traffic Class bits in packets originated by that upper-layer protocol. The default value must be zero for all 8 bits. Nodes that support a specific (experimental or eventual standard) use of some or all of the Traffic Class bits are permitted to change the value of those bits in packets that they originate, forward, or receive, as required for that specific use.
- o Nodes should ignore and leave unchanged any bits of the Traffic Class field for which they do not support a specific use.

- o An upper-layer protocol must not assume that the value of the Traffic Class bits in a received packet are the same as the value sent by the packet's source.

#### **N. MAXIMUM PACKET LIFETIME**

Unlike IPv4, IPv6 nodes are not required to enforce maximum packet lifetime. That is the reason the IPv4 "Time to Live" field was renamed "Hop Limit" in IPv6. In practice, very few, if any, IPv4 implementations conform to the requirement that they limit packet lifetime, so this is not a change in practice. Any upper-layer protocol that relies on the internet layer (whether IPv4 or IPv6) to limit packet lifetime ought to be upgraded to provide its own mechanisms for detecting and discarding obsolete packets.

#### **O. MAXIMUM UPPER-LAYER PAYLOAD SIZE**

When computing the maximum payload size available for upper-layer data, an upper-layer protocol must take into account the larger size of the IPv6 header relative to the IPv4 header. For example, in IPv4, TCP's MSS option is computed as the maximum packet size (a default value or a value learned through Path MTU Discovery) minus 40 octets (20 octets for the minimum-length IPv4 header and 20 octets for the minimum-length TCP header). When using TCP over IPv6, the MSS must be computed as the maximum packet size minus 60 octets, because the minimum-length IPv6 header (i.e., an IPv6 header with no extension headers) is 20 octets longer than a minimum-length IPv4 header.

## **P. SEMANTICS AND USAGE OF THE FLOW LABEL FIELD**

A flow is a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers. The nature of that special handling might be conveyed to the routers by a control protocol, such as a resource reservation protocol, or by information within the flow's packets themselves, e.g., in a hop-by-hop option. The details of such control protocols or options are beyond the scope of this document.

There may be multiple active flows from a source to a destination, as well as traffic that is not associated with any flow. A flow is uniquely identified by the combination of a source address and a non-zero flow label. Packets that do not belong to a flow carry a flow label of zero.

A flow label is assigned to a flow by the flow's source node. New flow labels must be chosen (pseudo-)randomly and uniformly from the range 1 to FFFFF hex. The purpose of the random allocation is to make any set of bits within the Flow Label field suitable for use as a hash key by routers, for looking up the state associated with the flow.

All packets belonging to the same flow must be sent with the same source address, destination address, and flow label. If any of those packets includes a Hop-by-Hop Options header, then they all must be originated with the same Hop-by-Hop Options header contents (excluding the Next Header field of the Hop-by-Hop Options header). If any of those packets includes a Routing header, then they all must be originated with the same contents in all extension headers up to and including the Routing header



(excluding the Next Header field in the Routing header). The routers or destinations are permitted, but not required, to verify that these conditions are satisfied. If a violation is detected, it should be reported to the source by an ICMP Parameter Problem message, Code 0, pointing to the high-order octet of the Flow Label field (i.e., offset 1 within the IPv6 packet).

The maximum lifetime of any flow-handling state established along a flow's path must be specified as part of the description of the state-establishment mechanism, e.g., the resource reservation protocol or the flow-setup hop-by-hop option. A source must not re-use a flow label for a new flow within the maximum lifetime of any flow-handling state that might have been established for the prior use of that flow label.

When a node stops and restarts (e.g., as a result of a "crash"), it must be careful not to use a flow label that it might have used for an earlier flow whose lifetime may not have expired yet. This may be accomplished by recording flow label usage on stable storage so that it can be remembered across crashes, or by refraining from using any flow labels until the maximum lifetime of any possible previously established flows has expired. If the minimum time for rebooting the node is known, that time can be deducted from the necessary waiting period before starting to allocate flow labels.

There is no requirement that all, or even most, packets belong to flows, i.e., carry non-zero flow labels. This observation is placed here to remind protocol designers and implementors not to assume otherwise. For example, it would be unwise to design a router whose performance would be adequate only if most packets belonged to flows, or

to design a header compression scheme that only worked on packets that belonged to flows.



## **APPENDIX D. TRANSPORT CONTROL PROTOCOL (TCP)**

This appendix provides Internet Standard Number 7, DoD Standard Transmission Control Protocol (TCP) (Information Sciences Institute 1981). The following are selected excerpts from Internet Request-For-Comments document 793.

---

### **A. INTRODUCTION**

The Transmission Control Protocol (TCP) is intended for use as a highly reliable host-to-host protocol between hosts in packet-switched computer communication networks, and in interconnected systems of such networks. This document describes the functions to be performed by the Transmission Control Protocol, the program that implements it, and its interface to programs or users that require its services.

### **B. MOTIVATION**

Computer communication systems are playing an increasingly important role in military, government, and civilian environments. This document focuses its attention primarily on military computer communication requirements, especially robustness in the presence of communication unreliability and availability in the presence of congestion, but many of these problems are found in the civilian and government sector as well.

As strategic and tactical computer communication networks are developed and deployed, it is essential to provide means of interconnecting them and to provide standard interprocess communication protocols which can support a broad range of applications. In anticipation of the need for such standards, the Deputy Undersecretary of Defense for

Research and Engineering has declared the Transmission Control Protocol (TCP) described herein to be a basis for DoD-wide inter-process communication protocol standardization.

TCP is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols which support multi-network applications. The TCP provides for reliable inter-process communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks. Very few assumptions are made as to the reliability of the communication protocols below the TCP layer. TCP assumes it can obtain a simple, potentially unreliable datagram service from the lower level protocols. In principle, the TCP should be able to operate above a wide spectrum of communication systems ranging from hard-wired connections to packet-switched or circuit-switched networks. The TCP fits into a layered protocol architecture just above a basic Internet Protocol which provides a way for the TCP to send and

### **C. INTERFACES**

The TCP interfaces on one side to user or application processes and on the other side to a lower level protocol such as Internet Protocol.

The interface between an application process and the TCP is illustrated in reasonable detail. This interface consists of a set of calls much like the calls an operating system provides to an application process for manipulating files. For example, there are calls to open and close connections and to send and receive data on established connections. It is also expected that the TCP can asynchronously communicate with

application programs. Although considerable freedom is permitted to TCP implementors to design interfaces which are appropriate to a particular operating system environment, a minimum functionality is required at the TCP/user interface for any valid implementation.

The interface between TCP and lower level protocol is essentially unspecified except that it is assumed there is a mechanism whereby the two levels can asynchronously pass information to each other. Typically, one expects the lower level protocol to specify this interface. TCP is designed to work in a very general environment of interconnected networks. The lower level protocol which is assumed throughout this document is the Internet Protocol

#### **D. OPERATION**

As noted above, the primary purpose of the TCP is to provide reliable, securable logical circuit or connection service between pairs of processes. To provide this service on top of a less reliable internet communication system requires facilities in the following areas:

- o Basic Data Transfer
- o Reliability
- o Flow Control
- o Multiplexing
- o Connections

- o Precedence and Security

The basic operation of the TCP in each of these areas is described in the following paragraphs.

1. **Basic Data Transfer:**

The TCP is able to transfer a continuous stream of octets in each direction between its users by packaging some number of octets into segments for transmission through the internet system. In general, the TCPs decide when to block and forward data at their own convenience.

Sometimes users need to be sure that all the data they have submitted to the TCP has been transmitted. For this purpose a push function is defined. To assure that data submitted to a TCP is actually transmitted the sending user indicates that it should be pushed through to the receiving user. A push causes the TCPs to promptly forward and deliver data up to that point to the receiver. The exact push point might not be visible to the receiving user and the push function does not supply a record boundary marker.

2. **Reliability:**

The TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the internet communication system. This is achieved by assigning a sequence number to each octet transmitted, and requiring a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. At the receiver, the sequence numbers are used to correctly order segments that may be received out of order and to eliminate duplicates. Damage is

handled by adding a checksum to each segment transmitted, checking it at the receiver, and discarding damaged segments.

As long as the TCPs continue to function properly and the internet system does not become completely partitioned, no transmission errors will affect the correct delivery of data. TCP recovers from internet communication system errors.

### **3. Flow Control:**

TCP provides a means for the receiver to govern the amount of data sent by the sender. This is achieved by returning a "window" with every ACK indicating a range of acceptable sequence numbers beyond the last segment successfully received. The window indicates an allowed number of octets that the sender may transmit before receiving further permission.

### **4. Multiplexing:**

To allow for many processes within a single Host to use TCP communication facilities simultaneously, the TCP provides a set of addresses or ports within each host. Concatenated with the network and host addresses from the internet communication layer, this forms a socket. A pair of sockets uniquely identifies each connection. That is, a socket may be simultaneously used in multiple connections.

The binding of ports to processes is handled independently by each Host. However, it proves useful to attach frequently used processes (e.g., a "logger" or timesharing service) to fixed sockets which are made known to the public. These services can then be accessed through the known addresses. Establishing and learning the port addresses of other processes may involve more dynamic mechanisms.



## **5. Connections:**

The reliability and flow control mechanisms described above require that TCPs initialize and maintain certain status information for each data stream. The combination of this information, including sockets, sequence numbers, and window sizes, is called a connection. Each connection is uniquely specified by a pair of sockets identifying its two sides. When two processes wish to communicate, their TCP's must first establish a connection (initialize the status information on each side). When their communication is complete, the connection is terminated or closed to free the resources for other uses.

Since connections must be established between unreliable hosts and over the unreliable internet communication system, a handshake mechanism with clock-based sequence numbers is used to avoid erroneous initialization of connections.

## **6. Precedence and Security:**

The users of TCP may indicate the security and precedence of their communication. Provision is made for default values to be used when these features are not needed.

## **E. MODEL OF OPERATION**

Processes transmit data by calling on the TCP and passing buffers of data as arguments. The TCP packages the data from these buffers into segments and calls on the internet module to transmit each segment to the destination TCP. The receiving TCP places the data from a segment into the receiving user's buffer and notifies the receiving user. The TCPs include control information in the segments which they use to ensure reliable ordered data transmission.

The model of internet communication is that there is an internet protocol module associated with each TCP which provides an interface to the local network. This internet module packages TCP segments inside internet datagrams and routes these datagrams to a destination internet module or intermediate gateway. To transmit the datagram through the local network, it is embedded in a local network packet.

The packet switches may perform further packaging, fragmentation, or other operations to achieve the delivery of the local packet to the destination internet module.

At a gateway between networks, the internet datagram is "unwrapped" from its local packet and examined to determine through which network the internet datagram should travel next. The internet datagram is then "wrapped" in a local packet suitable to the next network and routed to the next gateway, or to the final destination.

A gateway is permitted to break up an internet datagram into smaller internet datagram fragments if this is necessary for transmission through the next network. To do this, the gateway produces a set of internet datagrams; each carrying a fragment. Fragments may be further broken into smaller fragments at subsequent gateways. The internet datagram fragment format is designed so that the destination internet module can reassemble fragments into internet datagrams.

A destination internet module unwraps the segment from the datagram (after reassembling the datagram, if necessary) and passes it to the destination TCP.

This simple model of the operation glosses over many details. One important feature is the type of service. This provides information to the gateway (or internet module) to guide it in selecting the service parameters to be used in traversing the next

network. Included in the type of service information is the precedence of the datagram. Datagrams may also carry security information to permit host and gateways that operate in multilevel secure environments to properly segregate datagrams for security considerations.

## **F. THE HOST ENVIRONMENT**

The TCP is assumed to be a module in an operating system. The users access the TCP much like they would access the file system. The TCP may call on other operating system functions, for example, to manage data structures. The actual interface to the network is assumed to be controlled by a device driver module. The TCP does not call on the network device driver directly, but rather calls on the internet datagram protocol module which may in turn call on the device driver.

The mechanisms of TCP do not preclude implementation of the TCP in a front-end processor. However, in such an implementation, a host-to-front-end protocol must provide the functionality to support the type of TCP-user interface described in this document

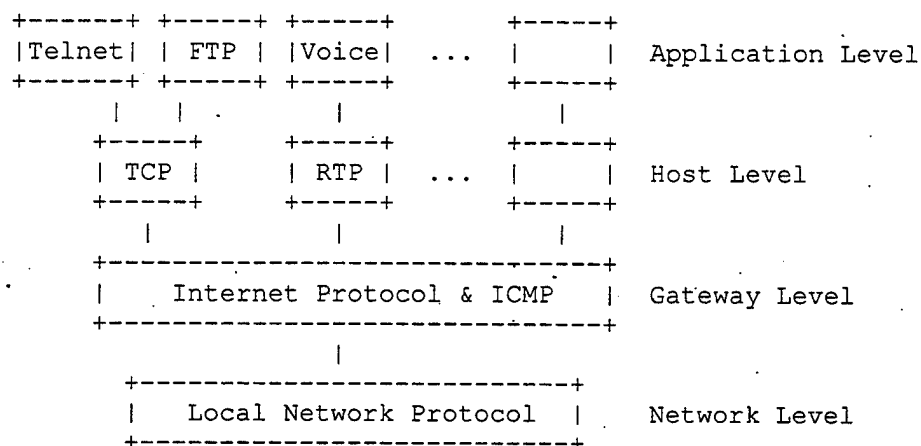
## **G. INTERFACES**

The TCP/user interface provides for calls made by the user on the TCP to OPEN or CLOSE a connection, to SEND or RECEIVE data, or to obtain STATUS about a connection. These calls are like other calls from user programs on the operating system, for example, the calls to open, read from, and close a file.

The TCP/internet interface provides calls to send and receive datagrams addressed to TCP modules in hosts anywhere in the internet system. These calls have parameters for passing the address, type of service, precedence, security, and other control information.

## H. RELATION TO OTHER PROTOCOLS

The following diagram illustrates the place of the TCP in the protocol hierarchy:



It is expected that the TCP will be able to support higher level protocols efficiently. It should be easy to interface higher level protocols like the ARPANET Telnet or AUTODIN II THP to the TCP.

## I. RELIABLE COMMUNICATION

A stream of data sent on a TCP connection is delivered reliably and in order at the destination. Transmission is made reliable via the use of sequence numbers and acknowledgments. Conceptually, each octet of data is assigned a sequence number. The

sequence number of the first octet of data in a segment is transmitted with that segment and is called the segment sequence number. Segments also carry an acknowledgment number which is the sequence number of the next expected data octet of transmissions in the reverse direction. When the TCP transmits a segment containing data, it puts a copy on a retransmission queue and starts a timer; when the acknowledgment for that data is received, the segment is deleted from the queue. If the acknowledgment is not received before the timer runs out, the segment is retransmitted.

An acknowledgment by TCP does not guarantee that the data has been delivered to the end user, but only that the receiving TCP has taken the responsibility to do so.

To govern the flow of data between TCPs, a flow control mechanism is employed. The receiving TCP reports a "window" to the sending TCP. This window specifies the number of octets, starting with the acknowledgment number, that the receiving TCP is currently prepared to receive.

## **J. CONNECTION ESTABLISHMENT AND CLEARING**

To identify the separate data streams that a TCP may handle, the TCP provides a port identifier. Since port identifiers are selected independently by each TCP they might not be unique. To provide for unique addresses within each TCP, we concatenate an internet address identifying the TCP with a port identifier to create a socket which will be unique throughout all networks connected together.

A connection is fully specified by the pair of sockets at the ends. A local socket may participate in many connections to different foreign sockets. A connection can be used to carry data in both directions, that is, it is "full duplex".

TCPs are free to associate ports with processes however they choose. However, several basic concepts are necessary in any implementation. There must be well-known sockets which the TCP associates only with the "appropriate" processes by some means. We envision that processes may "own" ports, and that processes can initiate connections only on the ports they own. (Means for implementing ownership is a local issue, but we envision a Request Port user command, or a method of uniquely allocating a group of ports to a given process, e.g., by associating the high order bits of a port name with a given process.). A connection is specified in the OPEN call by the local port and foreign socket arguments. In return, the TCP supplies a (short) local connection name by which the user refers to the connection in subsequent calls. There are several things that must be remembered about a connection. To store this information we imagine that there is a data structure called a Transmission Control Block (TCB). One implementation strategy would have the local connection name be a pointer to the TCB for this connection. The OPEN call also specifies whether the connection establishment is to be actively pursued, or to be passively waited for.

A passive OPEN request means that the process wants to accept incoming connection requests rather than attempting to initiate a connection. Often the process requesting a passive OPEN will accept a connection request from any caller. In this case a foreign socket of all zeros is used to denote an unspecified socket. Unspecified foreign sockets are allowed only on passive OPENs.

A service process that wished to provide services for unknown other processes would issue a passive OPEN request with an unspecified foreign socket. Then a

connection could be made with any process that requested a connection to this local socket. It would help if this local socket were known to be associated with this service.

Well-known sockets are a convenient mechanism for a priori associating a socket address with a standard service. For instance, the "Telnet-Server" process is permanently assigned to a particular socket, and other sockets are reserved for File Transfer, Remote Job Entry, Text Generator, Echoer, and Sink processes (the last three being for test purposes). A socket address might be reserved for access to a "Look-Up" service which would return the specific socket at which a newly created service would be provided. The concept of a well-known socket is part of the TCP specification, but the assignment of sockets to services is outside this specification.

Processes can issue passive OPENs and wait for matching active OPENs from other processes and be informed by the TCP when connections have been established. Two processes which issue active OPENs to each other at the same time will be correctly connected. This flexibility is critical for the support of distributed computing in which components act asynchronously with respect to each other.

There are two principal cases for matching the sockets in the local passive OPENs and an foreign active OPENs. In the first case, the local passive OPENs has fully specified the foreign socket. In this case, the match must be exact. In the second case, the local passive OPENs has left the foreign socket unspecified. In this case, any foreign socket is acceptable as long as the local sockets match. Other possibilities include partially restricted matches. If there are several pending passive OPENs (recorded in TCBs) with the same local socket, an foreign active OPEN will be matched to a TCB

with the specific foreign socket in the foreign active OPEN, if such a TCB exists, before selecting a TCB with an unspecified foreign socket.

The procedures to establish connections utilize the synchronize (SYN) control flag and involves an exchange of three messages. This exchange has been termed a three-way hand shake.

A connection is initiated by the rendezvous of an arriving segment containing a SYN and a waiting TCB entry each created by a user OPEN command. The matching of local and foreign sockets determines when a connection has been initiated. The connection becomes "established" when sequence numbers have been synchronized in both directions.

The clearing of a connection also involves the exchange of segments, in this case carrying the FIN control flag.

## **K. DATA COMMUNICATION**

The data that flows on a connection may be thought of as a stream of octets. The sending user indicates in each SEND call whether the data in that call (and any preceeding calls) should be immediately pushed through to the receiving user by the setting of the PUSH flag.

A sending TCP is allowed to collect data from the sending user and to send that data in segments at its own convenience, until the push function is signaled, then it must send all unsent data. When a receiving TCP sees the PUSH flag, it must not wait for more data from the sending TCP before passing the data to the receiving process.



There is no necessary relationship between push functions and segment boundaries. The data in any particular segment may be the result of a single SEND call, in whole or part, or of multiple SEND calls.

The purpose of push function and the PUSH flag is to push data through from the sending user to the receiving user. It does not provide a record service.

There is a coupling between the push function and the use of buffers of data that cross the TCP/user interface. Each time a PUSH flag is associated with data placed into the receiving user's buffer, the buffer is returned to the user for processing even if the buffer is not filled. If data arrives that fills the user's buffer before a PUSH is seen, the data is passed to the user in buffer size units.

TCP also provides a means to communicate to the receiver of data that at some point further along in the data stream than the receiver is currently reading there is urgent data. TCP does not attempt to define what the user specifically does upon being notified of pending urgent data, but the general notion is that the receiving process will take action to process the urgent data quickly.

## **L. PRECEDENCE AND SECURITY**

The TCP makes use of the internet protocol type of service field and security option to provide precedence and security on a per connection basis to TCP users. Not all TCP modules will necessarily function in a multilevel secure environment; some may be limited to unclassified use only, and others may operate at only one security level and compartment. Consequently, some TCP implementations and services to users may be limited to a subset of the multilevel secure case.

TCP modules which operate in a multilevel secure environment must properly mark outgoing segments with the security, compartment, and precedence. Such TCP modules must also provide to their users or higher level protocols such as Telnet or THP an interface to allow them to specify the desired security level, compartment, and precedence of connections.

#### **M. ROBUSTNESS PRINCIPLE**

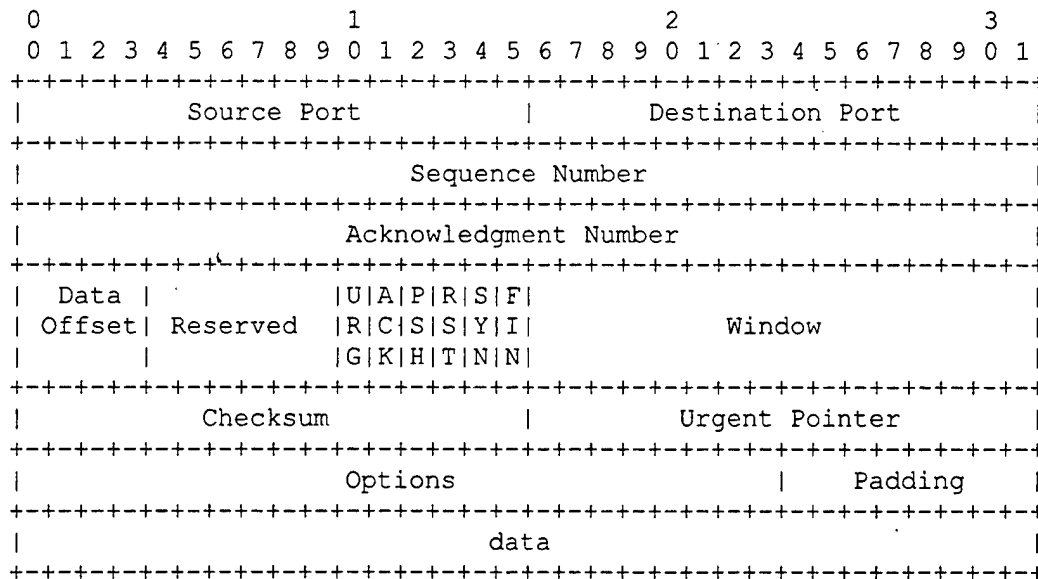
TCP implementations will follow a general principle of robustness: be conservative in what you do, be liberal in what you accept from others.

#### **N. FUNCTIONAL SPECIFICATION**

##### **1. Header Format**

TCP segments are sent as internet datagrams. The Internet Protocol header carries several information fields, including the source and destination host addresses. A TCP header follows the internet header, supplying information specific to the TCP protocol. This division allows for the existence of host level protocols other than TCP.

### TCP Header Format



### TCP Header Format

Note that one tick mark represents one bit position.

- o Source Port: 16 bits The source port number.
- o Destination Port: 16 bits The destination port number.
- o Sequence Number: 32 bits The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.
- o Acknowledgment Number: 32 bits If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

- o Data Offset: 4 bits The number of 32 bit words in the TCP Header.

This indicates where the data begins. The TCP header (even one including options) is an integral number of 32 bits long.

- o Reserved: 6 bits Reserved for future use. Must be zero.

- o Control Bits: 6 bits (from left to right):

URG: Urgent Pointer field significant

ACK: Acknowledgment field significant

PSH: Push Function

RST: Reset the connection

SYN: Synchronize sequence numbers

FIN: No more data from sender

- o Window: 16 bits The number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept.

- o Checksum: 16 bits The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header and text. If a segment contains an odd number of header and text octets to be checksummed, the last octet is padded on the right with zeros to form a 16 bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum, the checksum field itself is replaced with zeros. The checksum also covers a 96 bit pseudo header conceptually prefixed to the TCP header. This

pseudo header contains the Source Address, the Destination Address, the Protocol, and TCP length. This gives the TCP protection against misrouted segments. This information is carried in the Internet Protocol and is transferred across the TCP/Network interface in the arguments or results of calls by the TCP on the IP.

-----+			
	Source Address		
-----+			
	Destination Address		
-----+			
	zero	PTCL	
-----+			

The TCP Length is the TCP header length plus the data length in octets (this is not an explicitly transmitted quantity, but is computed), and it does not count the 12 octets of the pseudo header.

o Urgent Pointer:                    16 bits    This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field is only be interpreted in segments with the URG control bit set.

o Options:                                variable    Options may occupy space at the end of the TCP header and are a multiple of 8 bits in length. All options are included in the checksum. An option may begin on any octet boundary. There are two cases for the format of an option:

Case 1: A single octet of option-kind.

Case 2: An octet of option-kind, an octet of option-length, and the actual option-data octets. The option-length counts the two octets of option-kind and option-length as well as the option-data octets.

Note that the list of options may be shorter than the data offset field might imply. The content of the header beyond the End-of-Option option must be header padding (i.e., zero).

- o Padding:                      variable    The TCP header padding is used to ensure that the TCP header ends and data begins on a 32 bit boundary. The padding is composed of zeros.



## LIST OF REFERENCES

Cabletron (1997). Cabletron Systems ATM Technology Guide, Cabletron Systems, Inc.

Cisco (1996). ATM Primer. San Jose, Cisco Systems, Inc: 18.

Cisco (1997). White Paper: ATM Primer. San Jose, Cisco Systems, Inc: 10.

Cisco (1998). LightStream 2020 System Overview. San Jose, Cisco Systems. Inc.: 42.

Cisco (1998). Tag Switching: Uniting Routing and Switching for Scalable, High Performance Services. San Jose, Cisco Systems, Inc: 7.

Freier, Alan O., Philip Karlton, et al. (1996). The SSL Protocol Version 3.0, Internet Engineering Task Force: 65.

Harkins, D. and D. Carrel (1998). The Internet Key Exchange (IKE), Internet Engineering Task Force (IETF): 40.

Kent, Stephen and Randall Atkinson (1998). IP Authentication Header, Internet Engineering Task Force: 23.

Kent, Stephen and Randall Atkinson (1998). IP Encapsulating Security Payload, Internet Engineering Task Force: 22.

Kent, Stephen and Randall Atkinson (1998). Security Architecture for the Internet Protocol, Internet Engineering Task Force: 66.

Lauback, M. and J. Halpern (1998). Classical IP and ARP over ATM, Internet Engineering Task Force: 28.

Madson, C. and R. Glenn (1998). The Use of HMAC-MD5-96 Within ESP and AH, Internet Engineering Task Force: 6.



Madson, C. and R. Glenn (1998). The Use of HMAC-SHA-1-96 Within ESP and AH, Internet Engineering Task Force: 6.

Mitra, Suvo and Thomas Y.C. Woo (1997). "A Flow-Based Approach to Datagram Security." ACM: 221-234.

Perkins, C. (1996). Minimal Encapsulation within IP, Internet Engineering Task Force (IETF): 6.

Rechter, Yakov, Bruce Davie, et al. (1997). Tag Switching Architecture. San Jose, Cisco Systems.

Reynolds, J. and J. Postel (1994). Assigned Numbers, Internet Engineering Task Force: 230.

Stallings, William (1992). ISDN and Broadband ISDN. New York, Macmillan Publishing Company.

Stallings, William (1998). Cryptography and Network Security: Principles and Practice. Upper Saddle River, Prentice Hall, Inc.

Wu, Alan (1998). ATM Technology Overview, Mitre Corporation: 47.

## BIBLIOGRAPHY

- Bartee, T. C., N. W. Alvarez, et al. (1997). "Internet Security Label (ISL)", Internet Engineering Task Force (IETF): 1-18.
- Batson, M. (1997). "Tutorial and Reference: Asynchronous Transmission Mode (ATM)". Monterey, Naval Postgraduate School: 1-40.
- Bellovin, S. (1994). "Security Concerns for IPng", Internet Engineering Task Force (IETF): 1-4.
- Berinato, S., J. Kerstetter, et al. (1998). "VoIP Faces Major Hurdle: Limited security is hendering acceptance of voice over IP". PCWEEK. **15**: 18.
- Bitan, S. and D. Frommer (1997). "The Use of DES-MAC within ESP and AH", Internet Engineering Task Force: 1-7.
- Calhoun, P., G. Montenegro, et al. (1998). "Tunnel Establishment Protocol", Internet Engineering Task Force (IETF): 1-33.
- Cisco (1997). "Internetworking Technology Overview", CISCO Systems. **1998**.
- Clark, D. D. (1982). "IP Datagram Reassembly Algorithms", Internet Engineering Task Force (IETF): 1-9.
- Conta, A. and S. Deering (1998). "Generic Packet Tunneling in IPv6 Specification", Internet Engineering Task Force (IETF): 1-38.
- Deering, S. and R. Hinden (1997). "Internet Protocol, Version 6 (IPv6)", Internet Engineering Task Force (IETF): 1-41.
- Dennis, R. M. (1996, September). "Internetworking: Integrating IP/ATM LAN/WAN Security". Systems Management. Monterey, Naval Postgraduate School: 1-191.
- Fellows, J., J. Hemenway, et al. (1987). "The Architecture of a Distributed Trusted Computing Base". InProceedings. Washington, DC: 68-77.
- Fluckiger, F. (1995). Understanding Networked Multimedia: Applications and Technology. London, Prentice Hall.

Handel, R., M. N. Huber, et al. (1995). ATM Networks: Concepts, Protocols, Applications. Workingham, Addison-Wesley Publishing Company, Inc.

Housley, R. (1993). "Security Label Framework for the Internet", Internet Engineering Task Force (IETF): 1-14.

Information Sciences Institute, U. (1981). "Transmission Control Protocol: DARPA Internet Program Protocol Specification". Marina Del Rey, Internet Engineering Task Force: 1-85.

Institute, I. S. (1981). "Internet Protocol: DARPA Internet Program Protocol Specification", Internet Engineering Task Force (IETF): 1-125.

Maughan, D., M. Schertler, et al. (1998). "Internet Security Association and Key Management Protocol (ISAKMP)", Internet Engineering Task Force (IETF): 1-85.

McCann, J., S. Deering, et al. (1996). "Path MTU Discovery for IP Version 6", Internet Engineering Task Force (IETF): 1-15.

Metzger, P. and W. Simpson (1995). "IP Authentication Using Keyed MD5", Internet Engineering Task Force (IETF): 1-6.

Netscape "How SSL Works", Mountain View, Netscape Communications Corporation: 1-6.

Orman, H. K. "The OAKLEY Key Determination Protocol", Internet Engineering Task Force: 1-38.

Pereira, R., S. Anand, et al. (1998). "The ISAKMP Configuration Method", Internet Engineering Task Force (IETF): 1-12.

Perkins, C. (1996). "IP Encapsulation within IP", Internet Engineering Task Force: 1-14.

Piper, D. and D. Harkins (1998). "The Pre-Shared Key for the Internet Protocol", Internet Engineering Task Force (IETF): 1-5.

R.Pereira and P. Bhattacharya (1998). "IPsec Policy Data Model", Internet Engineering Task Force (IETF): 1-8.

Rescorla, E. (1998). "Diffie-Hellman Key Agreement Method", Internet Engineering Task Force (IETF): 1-7.

Russell, D. and S. G.T. Gangemi (1992). Computer Security Basics. Sebastopol, O'Reilly & Associates, Inc.

Simpson, W. (1995). "IP in IP Tunneling", Internet Engineering Task Force (IETF): 1-8.

Stallings, W. (1995). Network and Internetwork Security: Principles and Practice. Englewood Cliffs, Prentice Hall.

Stallings, W. (1997). Data and Computer Communications. Upper Saddle River, Prentice Hall.

Stevens, W. R. (1994). TCP/IP Illustrated, Volume 1: The Protocols. Reading, Addison-Wesley Publishing Company, Inc.

Stevens, W. R. (1996). TCP/IP Illustrated Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols. Reading, Addison-Wesley Publishing Company, Inc.

Stevens, W. R. and G. R. Wright (1995). TCP/IP Illustrated, Volume 2: The Implementation. Reading, Addison-Wesley Publishing Company, Inc.

Weissman, C. (1998). "BLACKER: Security for the DDN". IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, IEEE Computer Society Press.

Xie, G., C. Irvine, et al. (1998). "LLPF: An Architecture for Link Layer Packet Filtering". Unpublished Manuscript, Department of Computer Science, Naval Postgraduate School, September 1998.

Ziemba, G., D. Reed, et al. (1995). "Security Considerations for IP Fragment Filtering", Internet Engineering Task Force (IETF): 1-10.



## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center.....2  
8725 John J. Kingman Road, Ste. 0944  
Ft. Belvoir, Virginia 22060-6218
  
2. Dudley Knox Library.....2  
Naval Postgraduate School  
411 Dyer Rd.  
Monterey, California 93943-5101
  
3. Dr. Geoffrey Xie.....1  
Code CS/Xi  
Naval Postgraduate School  
Monterey, California 93943-5101
  
4. Dr. Cynthia Irvine.....1  
Code CS/Ir  
Naval Postgraduate School  
Monterey, California 93943-5101
  
5. Rex Buddenberg.....1  
Code SM/Bu  
Naval Postgraduate School  
Monterey, California 93943-5101
  
6. Gregorio G. Darroca.....2  
10341 Colony Park Drive  
Fairfax, Virginia 22032
  
7. Dr. Reuben Harris.....1  
Chair, Systems Management Department  
Code SM  
Naval Postgraduate School  
Monterey, California 93943-5101
  
8. Director, Naval Intelligence.....2  
4251 Suitland road  
Washington, D.C. 20395

9. Dr. Ted Lewis.....1  
Code CS/Le  
Naval Postgraduate School  
Monterey, California 93943-5101
10. Dr. G.M. Lundy.....1  
Code CS/Ln  
Naval Postgraduate School  
Monterey, California 93943-5101
11. Director, Marine Corps Research Center.....2  
MCCDC, Code: C40RC  
2040 Broadway Street  
Quantico, Virginia 22134-5107
12. Don Brutzman, Code UW/Br.....1  
Undersea Warfare Department  
Naval Postgraduate School  
Monterey, California 93943-5000
12. Dan Boger.....1  
Chairperson, Computer Science Department  
Naval Postgraduate School  
Monterey, California 93943-5000
13. Dr. Blaine Burnham.....1  
National Security Agency  
Research and Development Building  
R23  
9800 Savage Road  
Fort Meade, Maryland 20755-6000
14. CAPT Dan Galik.....1  
Space and Naval Warfare Systems Command  
PMW 161  
Building OT-1, Room 1024  
4301 Pacific Highway  
San Diego, California 92110-3127

15. Commander, Naval Security Group Command.....1  
Naval Security Group Headquarters  
9800 Savage Road  
Suite 6585  
Fort Meade, Maryland 20755-6585
16. Mr. George Bieber.....1  
Defense Information Systems Agency  
Center for Information Systems Security  
5113 Leesburg Pike, Suite 400  
Falls Church, Virginia 22041-3230
17. CDR Chris Perry.....1  
N643  
Presidential Tower 1  
2511 South Jefferson Davis Highway  
Arlington, Virginia 22202